



Conciencia Tecnológica

ISSN: 1405-5597

contec@mail.ita.mx

Instituto Tecnológico de Aguascalientes  
México

Alonso Amo, Fernando; Luna, Enrique  
Análisis Comparativo de Diferentes Enfoques para el Modelado Dinámico Orientado a Objetos  
Conciencia Tecnológica, núm. 14, agosto, 2000, pp. 26-33  
Instituto Tecnológico de Aguascalientes  
Aguascalientes, México

Disponible en: <http://www.redalyc.org/articulo.oa?id=94401406>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica  
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal  
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

# ANÁLISIS COMPARATIVO DE DIFERENTES ENFOQUES PARA EL MODELADO DINÁMICO ORIENTADO A OBJETOS

Fernando Alonso Amó  
falonso@fi.upm.es  
Facultad de Informática  
Universidad Politécnica de Madrid  
España

Enrique Luna Ramírez<sup>1</sup>  
fp22067@zipi.fi.upm.es  
Depto. Sistemas y Computación  
Instituto Tecnológico de Aguascalientes  
México

## RESUMEN

En el modelo orientado a objetos, el análisis del comportamiento dinámico de un sistema es fundamental para llevar a cabo su diseño. Sin embargo, dada la gran variedad de métodos que existen para el modelado dinámico, el diseñador que pretenda modelar dicho comportamiento se enfrentará a la dificultad de decidir que enfoque utilizar para realizar el diseño de su modelo. En base a esto, se hace necesario analizar y contrastar diferentes enfoques para determinar sus características más relevantes, ayudando así al diseñador a tomar una decisión al respecto. Un análisis realizado en este artículo sobre diferentes métodos mostró que tales métodos no presentan una diferencia significativa en términos de notación, sintaxis, y semántica, entre otras características; sin embargo, también se observó que éstos no describen en forma explícita cómo llevan a cabo algunas de las actividades involucradas en el proceso del modelado dinámico de un sistema. Así, se propuso un enfoque completo y explícito que describe por fases y actividades dicho proceso.

**Palabras clave:** modelo dinámico, estados, eventos, diagrama de estados, traza de eventos.

## 1. INTRODUCCIÓN

El modelado orientado a objetos constituye una forma diferente de pensar acerca de diversas situaciones en el mundo real. El proceso del desarrollo del modelo orientado a objetos se basa en el concepto de *objeto* que combina las estructuras de datos con los comportamientos en una entidad única. Lo mejor para modelar un sistema es examinar primero su estructura estática, esto es, la estructura de sus objetos y las relaciones entre ellos. Después, se examinan los cambios de los objetos y sus relaciones a través del tiempo. Los aspectos de un sistema relacionados con el

tiempo y sus cambios a través de éste constituyen lo que se conoce como el *modelo dinámico* [6]. Así entonces, el modelo orientado a objetos no sólo requiere de su estructura estática, sino también del modelado del comportamiento dinámico asociado a esta estructura para que la descripción de un sistema sea completa.

Existen diversos métodos para el modelado de un sistema, y particularmente, para el modelado de su comportamiento dinámico. Por lo tanto, si se desea utilizar alguno de tales métodos, es necesario conocer las características de cada uno de ellos para una selección apropiada.

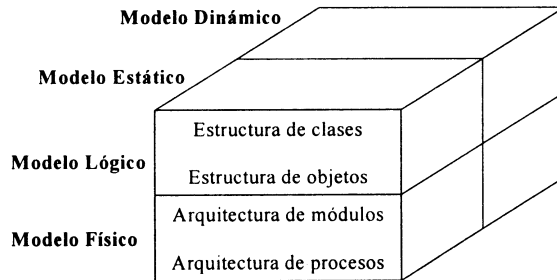
Este artículo está estructurado en seis secciones, partiendo con la actual introducción. En la sección dos se presentan los diferentes modelos que componen el modelo de objetos, como paso previo a la discusión del modelo dinámico en la sección tres. En la sección cuatro se discuten diferentes enfoques para el modelado dinámico de un sistema, y se hace un análisis comparativo de los mismos en base a un conjunto de aspectos tales como la notación, nivel de complejidad, nivel de detalle, mecanismo de partición, sintaxis y semántica definidas, automatización y ejemplos presentados en cada uno de ellos. Finalmente, en las secciones cinco y seis se presentan las conclusiones y referencias, respectivamente.

## 2. MODELOS DEL DESARROLLO ORIENTADO A OBJETOS

La tecnología orientada a objetos se apoya en los sólidos fundamentos de la ingeniería de software, cuyos elementos reciben el nombre global de *modelo de objetos*, que abarca los principios de abstracción, encapsulamiento, modularidad, jerarquía, tipos, concurrencia y persistencia. El enfoque del diseño orientado a objetos utiliza diversas notaciones para

<sup>1</sup> Doctorando del Instituto Tecnológico de Aguascalientes (México) en la Facultad de Informática de la Universidad Politécnica de Madrid. Este trabajo cuenta con el auspicio del Consejo Nacional de Ciencia y Tecnología de México (CONACYT).

expresar el modelo de objetos, y más concretamente, para expresar los diferentes modelos que lo componen. La figura 1 muestra la descomposición del modelo de objetos en sus diferentes partes: el modelo lógico, el modelo físico, y los modelos estático y dinámico [2].



**Figura 1:** Los modelos del desarrollo orientado a objetos [2].

Durante el análisis, debe definirse el comportamiento que se desea de un sistema y las misiones y responsabilidades de los objetos que llevan a cabo este comportamiento. El *modelo lógico* nos permite describir la existencia y el significado de las abstracciones principales (objetos y clases) y los mecanismos que forman el espacio del problema.

Durante el diseño, deben definirse los aspectos principales relativos a la arquitectura de un sistema: las clases que existen y como se relacionan entre sí, los mecanismos que se utilizan para regular la forma en que los objetos colaboran, el lugar donde debería declararse cada clase y objeto, a qué procesador debería asignarse cada proceso, y cómo deberían planificarse los múltiples procesos de cada procesador. En este sentido, el *modelo físico* nos permite describir la composición concreta en cuanto a hardware y software necesaria para la implantación del sistema.

Se pueden utilizar diferentes tipos de diagramas para definir la arquitectura de un sistema, entre los cuales se encuentran los diagramas de objetos, clases, módulos, y procesos. Estos diagramas forman el apoyo básico para el *modelo estático*, el cual describe la estructura de los objetos y las relaciones entre ellos. Sin embargo, los objetos tienen un comportamiento: en todos los sistemas basados en software, los objetos se crean y se destruyen, los objetos envían mensajes unos a otros de forma ordenada, y en algunos sistemas, eventos externos disparan operaciones sobre ciertos objetos. Este comportamiento es descrito por el *modelo dinámico*, el cual es tratado en la sección 3.

### 3. CONCEPTOS ASOCIADOS AL MODELO DINÁMICO

Cuando se considera el comportamiento dinámico de las cosas en el mundo real, observamos que esas cosas generalmente tienen un ciclo de vida. Con frecuencia algo que es creado, pasa a través de ciertas etapas de su existencia, y entonces muere o desaparece [7]. Considérense las siguientes observaciones generales en torno a los patrones de comportamiento para diferentes cosas en el mundo real:

- Muchas cosas pasan a través de varias etapas durante su ciclo de vida.
- El orden en el cual una cosa pasa a través de sus etapas forma un patrón característico de la clase de cosa que es.
- Una cosa en el mundo real está en una y sólo una de sus etapas de su patrón de comportamiento en un momento dado.
- Las cosas pasan de una etapa a otra en forma abrupta, en parte debido a la forma de definir o percibir las etapas.
- En un patrón de comportamiento, no todas las transiciones entre las diferentes etapas son permitidas. Algunas transiciones están prohibidas por las leyes de la física o por otras razones.
- Existen incidentes en el mundo real que causan el paso de las cosas de una etapa a otra.

Cuando se abstrae un grupo de “cosas” para crear una clase y posteriormente producir un objeto, también se está abstrayendo su patrón de comportamiento a través del ciclo de vida del objeto. Este ciclo provee una descripción formal del patrón de comportamiento formado por todas las instancias de una clase. Un ciclo de vida se puede expresar como un diagrama de transición de estados, es decir, la trama de estados, eventos y transiciones para una clase dada se puede abstraer y representar en forma de un diagrama de transición de estados. Así, dicho diagrama estará compuesto de:

- Estados. Cada estado representa una etapa en el ciclo de vida de un objeto.
- Eventos. Cada evento representa un incidente o la indicación de que está sucediendo una transición.
- Transiciones. Una transición específica a que nuevo estado debe pasar un objeto cuando recibe un evento particular.
- Acciones. Una acción es una actividad u operación que debe ser realizada cuando un objeto llega a un estado.

Considérese el siguiente ejemplo [7] para ilustrar los conceptos anteriores, los cuales serán tratados con mayor detalle cuando se discutan diferentes métodos para el modelado dinámico de un sistema.

**Ejemplo.** El proceso para la utilización de un horno de microondas involucra una serie de estados, eventos, transiciones y acciones descritos a continuación:

1. Existe un único botón de control disponible para el usuario del horno. Si la puerta del horno está cerrada y se oprime el botón, el horno comenzará a cocinar (es decir, se activa el tubo de poder) durante un minuto.
2. Si se oprime el botón en cualquier momento que el horno esté cocinando, se obtiene un minuto adicional de funcionamiento. Es decir, si se tienen 40 segundos más por cocinar y se oprime el botón dos veces, se tendrán automáticamente 2 minutos 40 segundos de cocinado.
3. Oprimir el botón cuando la puerta está abierta, no surte ningún efecto.
4. Existe una luz dentro del horno, la cual permanece encendida en todo momento que el horno esté cocinando. También, en cualquier momento que la puerta esté abierta, la luz permanecerá encendida.
5. Se puede detener el funcionamiento del horno abriendo la puerta.
6. Si se cierra la puerta, la luz se apaga. Esta es la situación normal cuando alguien ha colocado comida dentro del horno pero aún no ha presionado el botón de control.
7. Si el horno termina (llega al tiempo indicado de cocinado), se desactiva el tubo de poder y se apaga la luz, emitiéndose un sonido de alarma para indicar que el tiempo de cocción ha terminado.

En este problema existen varios incidentes que afectan el funcionamiento del horno, es decir, que producen cambios de estado en éste. Tales incidentes, que pueden ser abstraídos como eventos, son: “abrir la puerta”, “cerrar la puerta”, “oprimir el botón de control” y la “terminación del tiempo indicado para cocinar”. Por su parte, el horno puede cambiar de un estado a otro dentro de la siguiente gama de estados: “ocioso con la puerta cerrada”, “periodo inicial para cocinar”, “periodo extendido para cocinar”, “cocinado completo”, “ocioso con la puerta abierta”, y “cocinado interrumpido”. Así entonces, el funcionamiento del horno puede ser representado por un diagrama de transición de estados.

Cabe mencionar que los ciclos de vida de los diferentes objetos que intervienen en un sistema a menudo son coordinados entre sí. Por ejemplo, en el

caso del horno de microondas, el tubo de poder y la luz tienen ciclos de vida que están cercanamente ligados al estado en el cual se encuentra el horno. El diagrama de transición de estados del horno genera eventos de activación y desactivación que causan que el tubo de poder se mueva a través de su propio diagrama de transición de estados. De manera similar, el horno genera eventos que le indican a la luz encenderse o apagarse.

El ejemplo del horno de microondas tratado en esta sección ha sido de ayuda para comprender de manera intuitiva los conceptos e ideas asociadas al modelado dinámico de un sistema; sin embargo, tales conceptos requieren una descripción formal para poder utilizarlos en el análisis de los diferentes enfoques para el modelado dinámico que serán tratados en la siguiente sección.

### Estados, eventos y diagramas de transición de estados

El modelo estático describe los posibles conjuntos de objetos que pueden existir en un sistema, así como sus atributos y asociaciones. Los valores de los atributos y de las asociaciones mantenidas por un objeto forman lo que se denomina su *estado*. A lo largo del tiempo, los objetos actúan unos con otros, dando lugar a una serie de cambios en sus estados. Un estímulo individual de un objeto hacia otro se conoce con el nombre de *evento*. La respuesta de un objeto a un evento depende del estado en el cual se encuentra este, y puede conducir a un cambio de estado, o al envío de otro evento hacia el emisor o hacia un tercer objeto. Por su parte, un diagrama de transición de estados relaciona eventos y estados. Cuando se recibe un evento, el estado siguiente depende del actual, así como del evento mismo. Un cambio de estado causado por un evento es lo que se conoce como una *transición*.

Un *diagrama de transición de estados* es un grafo dirigido cuyos nodos representan los estados de un objeto, y cuyos arcos representan las posibles transiciones entre esos estados [7]. Los arcos se etiquetan con el nombre de los eventos correspondientes. Todas las transiciones que salgan de un estado deben de corresponder a eventos distintos. El diagrama de transición de estados especifica la secuencia de estados causada por una cierta secuencia de eventos. Si un objeto se encuentra en un cierto estado y se produce un evento cuyo nombre corresponda al de una de sus transiciones, entonces el objeto pasa al estado que se encuentra en el extremo de la transición; en este caso se dice que la transición *se dispara*. Si hay más de una transición que sale de un estado, entonces el primer evento que se produzca dará

lugar a que se dispare la transición correspondiente. Si se produce un evento que no tiene ninguna transición que salga del estado actual, entonces el evento se ignora. Así, una secuencia de eventos corresponde a un camino a través del grafo.

Un diagrama de transición de estados describe el comportamiento de una sola clase de objetos. Dado que todas las instancias de una clase (los objetos) tienen el mismo comportamiento, todas ellas comparten el mismo diagrama de transición de estados, ya que todas ellas comparten las mismas características de la clase. Pero dado que todo objeto posee sus propios valores de atributos, cada objeto posee su propio estado, que es el resultado de la secuencia particular de eventos que éste haya recibido.

#### Acciones y eventos versus factor tiempo

Una *acción* es una actividad u operación que debe ser ejecutada por una instancia a su llegada a un estado. Una acción es asociada a cada estado. Ya que todas las instancias de una clase comparten el mismo diagrama de estados, las acciones deben ser definidas de manera que puedan ser ejecutadas por cualquier instancia.

Una acción comúnmente lee y escribe atributos de la instancia para la cual se está ejecutando. Además, una acción puede:

- Hacer cualquier cálculo.
- Generar un evento para cualquier instancia de cualquier clase.
- Generar un evento para algo fuera del ámbito del análisis: un operador, un dispositivo de hardware, o un objeto en otro subsistema.
- Crear, borrar, establecer, restablecer o leer un temporizador<sup>2</sup>.
- Leer y escribir atributos de otras instancias de su propia clase.
- Leer y escribir atributos de las instancias de otras clases.

El formalismo obliga a hacer ciertas suposiciones acerca del tiempo con respecto a las acciones y los eventos:

<sup>2</sup> Un *temporizador* es un mecanismo que puede ser usado por una acción para generar un evento en algún momento futuro. Este puede ser visto como un objeto cuyos atributos son: un identificador para el temporizador mismo, la cantidad de tiempo que debe pasar antes de que el temporizador dispare, la etiqueta del evento que se generará cuando el temporizador dispare, y un identificador para el evento generado [7].

- Una única acción puede estar en ejecución en cualquier momento en un diagrama de estados dado.
- Acciones en diferentes diagramas de estado pueden ejecutarse simultáneamente.
- Si se genera un evento para una instancia que está ejecutando una acción, el evento no será aceptado hasta que la acción sea completada.
- Todo evento se agota completamente cuando se somete a un diagrama de estados: el evento desaparece como tal.
- Los eventos nunca se pierden, aunque pueden no ser tratados.

Los conceptos asociados al modelado dinámico de un sistema se aplican a lo largo de una serie de pasos que involucran notaciones que dependerán del enfoque utilizado por cada autor. En este sentido, en la siguiente sección se analizarán las propuestas de diferentes autores para el modelado dinámico.

#### 4. ALGUNOS MÉTODOS PARA EL MODELADO DINÁMICO

En esta sección se describen las características sobresalientes de algunos métodos propuestos por diferentes autores para el modelado dinámico de un sistema. Se hace una comparación de tales métodos en base a un conjunto de aspectos descritos al final de la sección.

##### Método de Booch

Booch [2] expresa la semántica dinámica de un problema o su implantación mediante dos diagramas:

- Diagramas de transición de estados
- Diagramas de interacción

Un diagrama de transición de estados se utiliza para mostrar el espacio de estados de una clase determinada, los eventos que causan la transición de un estado a otro, y las acciones que resultan de ese cambio de estado. Cada clase puede tener un diagrama de transición de estados que indica el comportamiento de las instancias de esa clase de cara al orden de sucesión de los eventos. De la misma forma, en conjunción con un escenario<sup>3</sup>, se puede utilizar un

<sup>3</sup> Un *escenario* es una secuencia de eventos que se produce durante una ejecución concreta de un sistema. El ámbito de un escenario es variable; puede incluir a todos los eventos del sistema, o puede incluir solamente aquellos eventos que afectan a ciertos objetos del sistema, o que sean generados por ellos.

diagrama de interacción para mostrar el orden temporal en que se evalúan los eventos.

Como se mencionó en la sección tres, un diagrama de transición de estados es un grafo dirigido cuyos nodos representan a los estados y arcos a los eventos, que surgen del análisis del comportamiento de los objetos de un sistema. Booch (y también Rumbaugh, como se comentará más adelante) utiliza rectángulos con esquinas redondeadas para representar a los estados, y arcos rotulados para representar a los eventos. Cada arco se rotula tanto con el nombre del evento que representa como con el nombre de la acción que debe ejecutarse una vez que el evento llegue a su estado de destino. Así entonces, cada arco tiene un rótulo de *evento/acción*. En la figura 2 se muestra el diagrama de transición de estados para el caso del horno de microondas descrito en la sección 3. Cabe señalar que Booch utiliza además un icono especial en sus diagramas de transición de estados, en forma de círculo relleno, para indicar la existencia de un estado inicial.

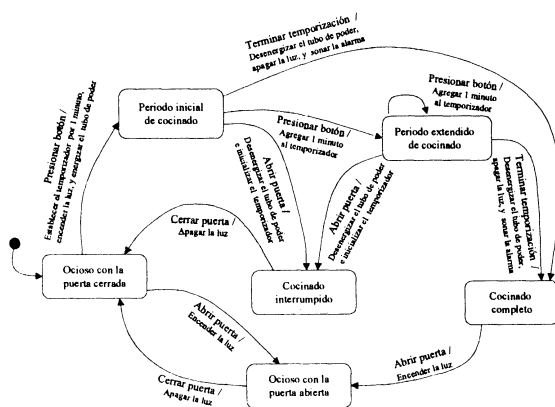


Figura 2: Diagrama de transición de estados de Booch para el caso del horno de microondas.

Respecto a los diagramas de interacción, estos son usados por Booch para realizar la traza de ejecución de un escenario. Estos diagramas son análogos a las *trazas de eventos* utilizadas por Rumbaugh et al. [6] con el mismo propósito. Así, dado que Rumbaugh et al. describen con mayor detalle la traza de ejecución de un escenario, tal descripción se realizará cuando se discuta su método.

### Método de Martin & Odell

Martin & Odell [5] utilizan diversos tipos de diagramas para modelar el comportamiento dinámico de un sistema:

- Diagramas de transición de estados

- Diagramas de interacción
- Diagramas de eventos
- Diagramas de flujo de objetos
- Diagramas de flujo de datos

Al igual que en los demás métodos, un *diagrama de transición de estados* expresa los diversos estados de un objeto, sus transiciones permisibles, y sus eventos y acciones asociadas.

Un cambio de estado en un objeto puede disparar un cambio de estado en otro objeto; sin embargo, representar todos los diagramas de transición y sus interacciones en un mismo diagrama complicaría su uso. Martin & Odell proponen el uso de *diagramas de interacción* para representar todos y cada uno de los diagramas de transición de estados por medio de sus objetos. Sugieren el uso de los diagramas de interacción de Coad (véase [5]) y de Jacobson [4], entre otros.

Los *diagramas de eventos* son descritos usando cuatro conceptos básicos: operaciones, eventos, disparadores, y condiciones de control. Las operaciones son procesos que llevan a cabo los cambios de estado. Los tipos de eventos definen los cambios de estado que resultan de las operaciones e invocan otras operaciones por medio de disparadores (reglas de disparo). Las condiciones de control aseguran que un cierto estado exista antes de que una operación pueda ser disparada sobre este.

Martin & Odell hacen una extensa clasificación de los eventos de acuerdo al tipo de cambios de estado que se producen en los objetos: agregar o remover un objeto, o una relación. Definen los conceptos de evento de creación, terminación, clasificación, desclasificación, conexión, desconexión, reclasificación, y reconexión; definen además los conceptos de pre-estados y post-estados, eventos internos, eventos externos, eventos temporales, y particiones de eventos, entre otros. Los autores presentan la notación necesaria para la representación de todos estos conceptos.

Los diagramas de interacción son de ayuda para determinar el ámbito de sistemas pequeños o medianos (caso del horno de microondas). Sin embargo, los sistemas grandes requieren de otra técnica. Los *diagramas de flujo de objetos* son utilizados para este propósito. Se llama flujo de objetos porque modela la forma en que los objetos son producidos por una actividad y consumidos por otra. En otras palabras, describe la forma en que los objetos fluyen de una actividad a otra.

Los *diagramas de flujo de datos* son una forma de representar la especificación funcional. Aunque los diagramas de flujo de datos no están pensados exclusivamente para la orientación a objetos, pueden ser modificados para incorporar una filosofía orientada a objetos.

**Método de Rumbaugh et al.**

Rumbaugh et al. [6] describen conceptos relacionados con el flujo de control, las interacciones y el secuenciado de las operaciones de un sistema de objetos concurrentemente activos. Los conceptos más importantes de su enfoque para el modelado dinámico son los eventos, que representan estímulos externos, y los estados, que representan los valores de los objetos. Para describir el comportamiento dinámico de los objetos de un sistema, este método utiliza la siguiente notación:

- Escenarios
- Trazas de eventos
- Diagramas de estados
- Diagramas de estados anidados
- Diagramas de estados concurrentes
- Diagramas de estados extendidos

El modelado dinámico comienza con la presentación de un *escenario* que muestra la lista de los eventos que intervienen en una determinada parte del sistema en cuestión. En la tabla 1 se muestra el escenario correspondiente al caso del horno de microondas.

|                              |
|------------------------------|
| Botón presionado             |
| Enciende la luz              |
| Energiza el tubo de poder    |
| Temporizador termina         |
| Apaga la luz                 |
| Desenergiza el tubo de poder |
| Puerta abierta               |
| Puerta cerrada               |

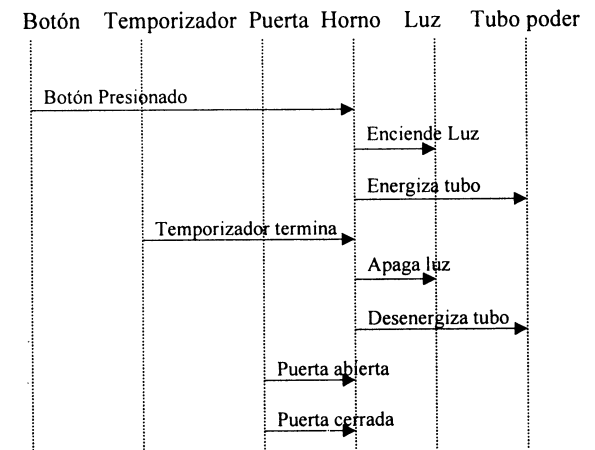
**Tabla 1:** Escenario de Rumbaugh et al. para el caso del horno de microondas.

Sobre este escenario se identifican los objetos emisores y receptores de cada evento. La secuencia de eventos y los objetos que intercambian tales eventos se pueden mostrar ambos en un diagrama (escenario mejorado) denominado *traza de eventos*. Este diagrama muestra cada objeto como una línea vertical y cada evento como una línea horizontal que va del objeto emisor al objeto receptor. El tiempo aumenta de arriba hacia abajo, pero el espaciado es irrelevante; lo único que se muestra son las secuencias de eventos, y no su temporización. En la figura 3 se muestra la traza de eventos correspondiente al escenario de la tabla 2.

De un diagrama de traza de eventos se pueden derivar los diagramas de estados para los diferentes objetos que intervienen en el análisis. En el caso del horno de microondas, se pueden derivar los diagramas de estados para el horno mismo, así como para el tubo de poder y la luz. En particular, el diagrama de estados para el horno de microondas es el mismo diagrama que utiliza Booch (mostrado en la figura 2), con la única diferencia que este último utiliza un icono de estado inicial.

En el modelado dinámico pueden surgir situaciones que presentan características de anidamiento de actividades, generalización de estados y eventos, concurrencia de agregación, concurrencia dentro de un objeto, y sincronización de actividades concurrentes, para lo cual los autores proporcionan la notación necesaria en base a una especialización de los diagramas de estados (diagramas anidados, concurrentes, y extendidos). Esto permite llevar a cabo el análisis del comportamiento de un sistema mediante refinamientos, es decir, ir de lo particular a lo general.

**Figura 3:** Diagrama de traza de eventos de Rumbaugh



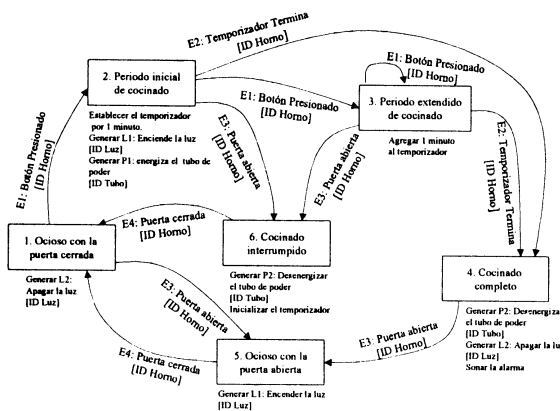
para el caso del horno de microondas.

**Método de Shlaer & Mellor**

Shlaer & Mellor [7] describen el comportamiento dinámico de los objetos en términos de *modelos de estados*. Un modelo de estados inicialmente se propone como un diagrama compuesto por un conjunto de estados, un conjunto de eventos, reglas de transición, y acciones. A tal diagrama le denominan *diagrama de transición de estados*. El concepto de modelo de estados es complementado con una *tabla de transición de estados*, que es una forma tabular del concepto de diagrama de transición de estados. Otros conceptos que utiliza este método son las listas de eventos y los diagramas de flujo de datos para las acciones. Una *lista de eventos* es un listado de todos

los eventos que intervienen en un sistema dado. Un *diagrama de flujo de datos para una acción* provee una representación gráfica de las unidades de procesamiento en una acción y de la intercomunicación entre ellas. Así entonces, los componentes de la notación de Shlaer & Mellor son:

- Diagramas de transición de estados
- Tablas de transición de estados
- Listas de eventos
- Diagramas de flujo de datos para una acción



**Figura 4:** Diagrama de transición de estados para el caso del horno de microondas bajo la notación de Shlaer & Mellor.

En la figura 4 se muestra nuevamente el diagrama de transición de estados para el caso del horno de microondas bajo la notación de este método. Los estados se representan por medio de rectángulos, cada uno etiquetado con un nombre apropiado para el estado correspondiente. Las transiciones se muestran como arcos que conectan pares de estados. Cada transición está etiquetada con el evento que causa la transición. La acción asociada con un estado es descrita bajo el rectángulo correspondiente. Además, se indica el objeto de destino de un evento dado mediante un identificador del objeto: ID <identificador del objeto>.

| ESTADO                            | EVENTO | Botón Pres. | Tempor. Termina | Puerta Abierta | Puerta Cerr. |
|-----------------------------------|--------|-------------|-----------------|----------------|--------------|
| 1. Ocio con la puerta cerrada     |        | 2           | NPS [1]         | 5              | NPS[3]       |
| 2. Periodo inicial para cocinar   |        | 3           | 4               | 6              | NPS[3]       |
| 3. Periodo extendido para cocinar |        | 3           | 4               | 6              | NPS[3]       |
| 4. Cocinado completo              |        | EI          | NPS[1]          | 5              | NPS[3]       |
| 5. Ocio con la puerta abierta     |        | EI          | NPS[1]          | NPS[2]         | 1            |
| 6. Cocinado interrumpido          |        | EI          | NPS[1]          | NPS[2]         | 1            |

[1] El temporizador no está funcionando [2]La puerta ya está abierta

[3] La puerta ya está cerrada  
EI: Evento Ignorado NPS: No Puede Suceder

**Tabla 2:** Tabla de transición de estados de Shlaer & Mellor para el caso del horno de microondas.

La tabla 2 muestra la tabla de transición de estados correspondiente al diagrama de la figura 4. En esta tabla, cada fila representa uno de los posibles estados del objeto (horno), y cada columna representa un evento que tiene a este objeto como su destino. Las celdas de la tabla contienen información que especifica lo que sucede cuando un estado (fila) dado recibe un particular evento (columna). La información contenida en las celdas puede ser de tres tipos: “estado nuevo”, “evento ignorado”, y “no puede suceder”.

### Análisis comparativo de los métodos

En la tabla 3 se presenta una comparación de los métodos descritos anteriormente en base a un conjunto de aspectos tales como la notación, nivel de complejidad, nivel de detalle, mecanismo de partición, sintaxis y semántica definidas, automatización, y ejemplos presentados.

### 5. CONCLUSIONES

- En general, los métodos tratados en este artículo no presentan diferencias significativas con referencia a los aspectos descritos en la Tabla 3 (notación, nivel de complejidad, nivel de detalle...).
- En particular, la sintaxis y semántica están suficientemente definidas haciendo factible la automatización de los métodos.
- El método de Rumbaugh et al. realiza además del refinamiento sucesivo top-down, anidamiento y especialización sobre los diagramas de transición de estados.

### 6. REFERENCIAS

- 1) Berard, E.V. “A Comparison of Object-Oriented Methodologies”, <http://www.toa.com/>, visitado el 14/Febrero/2000.
- 2) Booch, G. “Análisis y Diseño Orientado a Objetos con aplicaciones”, Segunda Edición, Addison-Wesley, 1996.
- 3) De Champeaux, D. & Faure, P. “A comparative study of object-oriented analysis methods”, *JOOP*, March/April 1992, pp. 21-33.



- 4) Jacobson, I. "Object-Oriented Software Engineering: A Use Case Driven Approach", Addison-Wesley, 1996.
- 5) Martin, J. and Odell, J.J. "Object-Oriented Methods: Pragmatic Considerations", Prentice Hall PTR, 1996.
- 6) Rumbaugh, J. et al. "Modelado y diseño orientado a objetos", Prentice Hall, 1996.
- 7) Shlaer, S. & Mellor, S.J. "Object Lifecycles: Modeling the World in States", Yourdon Press PTR Prentice Hall, 1992.

| Método          | Notación   | Nivel de complejidad | Nivel de detalle | Mecanismo de partición  | Sintaxis y Semántica definidas | Automatizable | Provee ejemplos |
|-----------------|--|----------------------|------------------|---|--------------------------------|---------------|-----------------|
| Booch           | <ul style="list-style-type: none"> <li>Diagramas de transición de estados</li> <li>Diagramas de interacción</li> </ul>   | Medio                | Medio            | <ul style="list-style-type: none"> <li>Refinamiento sucesivo top-down</li> </ul>  | Total                          | Todo          | Algunos         |
| Martin & Odell  | <ul style="list-style-type: none"> <li>Diagramas de transición de estados</li> <li>Diagramas de interacción</li> <li>Diagramas de eventos</li> <li>Diagramas de flujo de objetos</li> <li>Diagramas de flujo de datos</li> </ul>                 | Alto                 | Medio            | <ul style="list-style-type: none"> <li>Refinamiento sucesivo top-down</li> </ul>  | Total                          | Todo          | Algunos         |
| Rumbaugh        | <ul style="list-style-type: none"> <li>Escenarios</li> <li>Trazas de eventos</li> <li>Diagramas de estados</li> <li>Diagramas de estados anidados</li> <li>Diagramas de estados concurrentes</li> <li>Diagramas de estados extendidos</li> </ul> | Medio                | Alto             | <ul style="list-style-type: none"> <li>Refinamiento sucesivo top-down</li> <li>Anidamiento y especialización de diagramas de estados</li> </ul> | Parcial                        | Parte         | Muchos          |
| Shlaer & Mellor | <ul style="list-style-type: none"> <li>Modelos de estados</li> <li>Diagramas de transición de estados</li> <li>Tablas de transición de estados</li> <li>Listas de eventos</li> <li>Diagramas de flujo de datos para una acción</li> </ul>        | Medio                | Alto             | <ul style="list-style-type: none"> <li>Refinamiento sucesivo top-down</li> </ul>  | Total                          | Todo          | Muchos          |

Tabla 3: Análisis comparativo de los métodos.

