Programación dirigida por eventos

La **programación dirigida por eventos**, es un paradigma de programación en el que tanto la estructura como la ejecución de los programas van determinados por los sucesos que ocurran en el sistema, definidos por el usuario o que ellos mismos provoquen.

Para entender la programación dirigida por eventos, podemos oponerla a lo que no es: mientras en la programación secuencial (o estructurada) es el programador el que define cuál va a ser el flujo del programa, en la programación dirigida por eventos será el propio usuario —o lo que sea que esté accionando el programa— el que dirija el flujo del programa. Aunque en la programación secuencial puede haber intervención de un agente externo al programa, estas intervenciones ocurrirán cuando el programador lo haya determinado, y no en cualquier momento como puede ser en el caso de la programación dirigida por eventos.

El creador de un programa dirigido por eventos debe definir los eventos que manejarán su programa y las acciones que se realizarán al producirse cada uno de ellos, lo que se conoce como el administrador de evento. Los eventos soportados estarán determinados por el lenguaje de programación utilizado, por el sistema operativo e incluso por eventos creados por el mismo programador.

En la programación dirigida por eventos, al comenzar la ejecución del programa se llevarán a cabo las inicializaciones y demás código inicial y a continuación el programa quedará bloqueado hasta que se produzca algún evento. Cuando alguno de los eventos esperados por el programa tenga lugar, el programa pasará a ejecutar el código del correspondiente administrador de evento. Por ejemplo, si el evento consiste en que el usuario ha hecho clic en el botón de play de un reproductor de películas, se ejecutará el código del administrador de evento, que será el que haga que la película se muestre por pantalla.

Un ejemplo claro lo tenemos en los sistemas de programación Léxico y Visual Basic, en los que a cada elemento del programa (objetos, controles, etcétera) se le asignan una serie de eventos que generará dicho elemento, como la pulsación de un botón del ratón sobre él o el redibujado del control. O en Javascript que asigna manejadores de eventos a los que responder a eventos en una web en el caso del navegador o a eventos producidos por objetos emisores en el caso de NodeJS.

La programación dirigida por eventos es la base de lo que llamamos interfaz de usuario, aunque puede emplearse también para desarrollar interfaces entre componentes de Software o módulos del núcleo.

En los primeros tiempos de la computación, los programas eran secuenciales, también llamados Batch. Un programa secuencial arranca, lee parámetros de entrada, procesa estos parámetros, y produce un resultado, todo de manera lineal y sin intervención del usuario mientras se ejecuta.

Con la aparición y popularización de los PC, el software empezó a ser demandado para usos alejados de los clásicos académicos y empresariales para los cuales era necesitado hasta entonces, y quedó patente que el paradigma clásico de programación no podía responder a las nuevas necesidades de interacción con el usuario que surgieron a raíz de este hecho.

Índice

Detección de eventos

En contraposición al modelo clásico, la programación orientada a eventos permite interactuar con el usuario en cualquier momento de la ejecución. Esto se consigue debido a que los programas creados bajo esta arquitectura se componen por un bucle exterior permanente encargado de recoger los eventos, y distintos procesos que se encargan de tratarlos. Habitualmente, este bucle externo permanece oculto al programador que simplemente se encarga de tratar los eventos, aunque en algunos entornos de desarrollo (IDE) será necesaria su construcción.

Ejemplo de programa orientado a eventos en pseudo lenguaje:

```
While (true){
      Switch (event){
          case mouse_button_down:
          case mouse_click:
          case keypressed:
          case Else:
      }
  }
```

Problema

La programación orientada a eventos supone una complicación añadida con respecto a otros paradigmas de programación, debido a que el flujo de ejecución del software escapa al control del programador. En cierta manera podríamos decir que en la programación clásica el flujo estaba en poder del programador y era este quien decidía el orden de ejecución de los procesos, mientras que en programación orientada a eventos, es el usuario el que controla el flujo y decide.

Pongamos como ejemplo de la problemática existente, un menú con dos botones, botón 1 y botón 2. Cuando el usuario pulsa botón 1, el programa se encarga de recoger ciertos parámetros que están almacenados en un fichero y calcular algunas variables. Cuando el usuario pulsa el botón 2, se le muestran al usuario por pantalla dichas variables. Es sencillo darse cuenta de que la naturaleza indeterminada de las acciones del usuario y las características de este paradigma pueden fácilmente desembocar en el error fatal

de que se pulse el botón 2 sin previamente haber sido pulsado el botón 1. Aunque esto no pasa si se tienen en cuenta las propiedades de dichos botones, haciendo inaccesible la pulsación sobre el botón 2 hasta que previamente se haya pulsado el botón 1.

GUI's / Interfaces Gráficas de Usuarios

Con la evolución de los lenguajes orientados a eventos, la interacción del software con el usuario ha mejorado enormemente permitiendo la aparición de interfaces que, aparte de ser la vía de comunicación del programa con el usuario, son la propia apariencia del mismo. Estas interfaces, también llamadas GUI (Graphical User Interface), han sido la herramienta imprescindible para acercar la informática a los usuarios, permitiendo en muchos casos, a principiantes utilizar de manera intuitiva y sin necesidad de grandes conocimientos, el software que ha colaborado a mejorar la productividad en muchas tareas.

Uno de los periféricos que ha cobrado mayor importancia tras la aparición de los programas orientados a eventos ha sido el ratón, gracias también en parte a la aparición de los sistemas operativos modernos con sus interfaces gráficas. Estas suelen dirigir directamente al controlador interior que va entrelazado al algoritmo.

Herramientas visuales de desarrollo

Con el paso del tiempo, han ido apareciendo una nueva generación de herramientas que incluyen código que automatiza parte de las tareas más comunes en la detección y tratamiento de eventos.

Destacan particularmente los entornos de programación visual que conjugan una herramienta de diseño gráfica para la GUI y un lenguaje de alto nivel. Entre estas herramientas se encuentra la conocida Visual Basic, lenguaje altamente apreciado por principiantes debido a la facilidad para desarrollar software en poco tiempo y con pocos conocimientos, y denostado por tantos otros debido a su falta de eficiencia.

Lenguajes

Web

- ActionScript

Escritorio Windows

- Visual Basic
- Visual Object
- Visual C++
- Visual C#

.NET Framework (Escritorio Windows y Web)

- Visual Basic .NET
- C#
- J#
- Léxico

Otros

- NesC[1]
- AS3

Bibliotecas

C y C++

- Qt
- GTK+

Java

- AWT
- Swing
- SWT
- JavaFX

Web

- ASP.NET (Mediante Javascript con el Modelo Code-behind)

Véase también

- Programación estructurada
- Paradigma de programación
- Programación lógica

Referencias

- Grant Palmer: *Java Event Handling*, Prentice Hall, ISBN 0-13-041802-1.
- David Luckham: *The Power of Events - An Introduction to Complex Event Processing in Distributed Enterprise Systems*, Addison-Wesley, ISBN 0-201-72789-7.
- George S. Fishman: *Discrete-Event Simulation - Modeling, Programming, and Analysis*, Springer, ISBN 0-387-95160-1.
- Bertrand Meyer (2004): *The power of abstraction, reuse and simplicity: an object-oriented library for event-driven design*, in *Festschrift in Honor of Ole-Johan Dahl*, eds. Olaf Owe et al., Springer-Verlag, Lecture Notes in Computer Science 2635, en línea (http://se.ethz.ch/~meyer/publications/lncs/events.pdf).
- Miro Samek: *Practical Statecharts in C/C++: Quantum Programming for Embedded Systems*, CMP Books, ISBN 1-57820-110-1.
- Faison, Ted (2006). *Event-Based Programming: Taking Events to the Limit*. Apress. ISBN 1-59059-643-9.
- Adolfo Lozano Tello: *Iniciación a la programación utilizando lenguajes visuales orientados a eventos*, Ed.Bellisco Ediciones Técnicas y Científicas, ISBN 84-95279-49-5. ISBN 978-84-

95279-49-1

1. «NesC» (https://es.wikipedia.org/w/index.php?title=NesC&oldid=68274472) |url= incorrecta con autorreferencia (ayuda). *Wikipedia, la enciclopedia libre*. 11 de julio de 2013. Consultado el 8 de junio de 2016.

Enlaces externos

- Description (http://c2.com/cgi/wiki?EventDrivenProgramming) from Portland Pattern Repository

---