

Invocación Implícita

Se considera como una técnica de integración.

Los componentes son módulos cuyas interfaces proveen una colección de procedimientos y un conjunto de eventos. Los procedimientos se llaman de la manera usual pero el componente también puede activar algunos de sus procedimientos con los eventos del sistema. Esto hará que estos procedimientos sean invocados cuando los eventos ocurren en tiempo de ejecución.

Los generadores de eventos no saben cuáles componentes se afectarán por el evento. Ejemplos de este estilo son los sistemas de gestión de bases de datos cuando aseguran la consistencia de los datos, las aplicaciones con interfaces de usuarios al separar la representación de los datos de las aplicaciones que las gerencian.

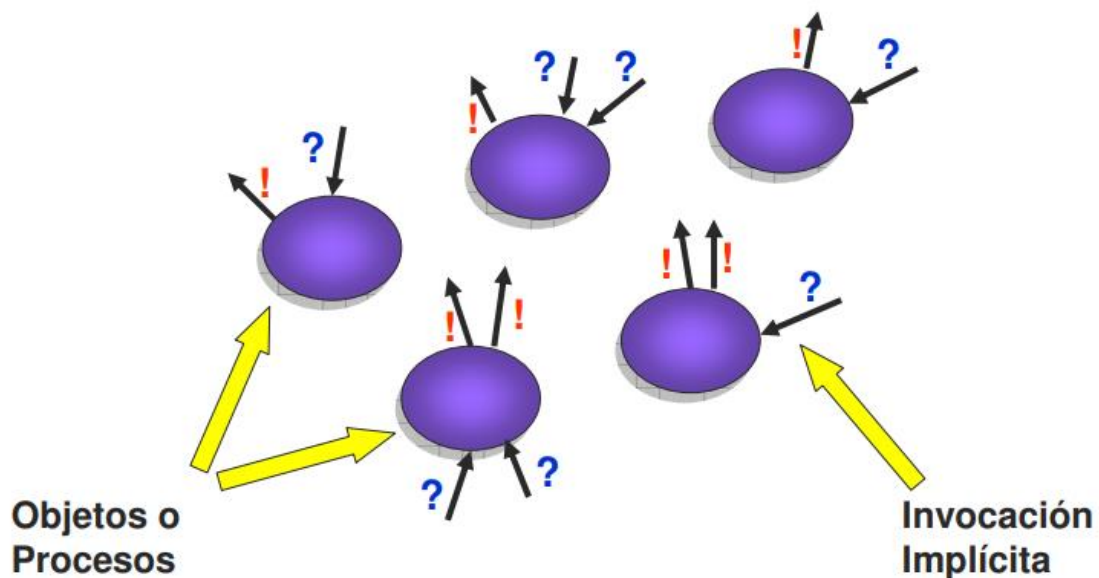
En lugar de invocaciones de procedimientos explícitas o directas, un componente anuncia uno o más eventos y otros componentes registran el interés en un evento asociando un procedimiento a dicho evento.

La ocurrencia de un evento causa la invocación “implícita” de procedimientos en otros módulos.

Los componentes son los módulos cuyas interfaces ofrecen un conjunto de procedimientos y de eventos.

Los conectores incluyen llamadas a procedimientos tradicionales, así como el ligado de eventos con llamadas a procedimientos.

La arquitectura con tareas de control utilizando invocación implícita (TCA) se basa en jerarquías de tareas, es decir, se generan árboles de tareas (tareas padre inician hijas con restricciones temporales)



Características:

- Se registran procedimientos para los eventos.
- Un componente comunica un evento.
- Cuando se anuncia un evento los procedimientos asociados son invocados implícitamente.
- El orden de invocación es no determinista.

Restricciones:

- Quien anuncia el evento no conoce a que componentes afecta el evento.
- No se pueden hacer asunciones acerca del orden de procesamiento.

Ventajas:

- Provee un robusto soporte de reusabilidad.
- Facilita la evolución del sistema
- Fácil cambiar los componentes que atienden un evento

Desventajas:

- Pérdida de control en el comportamiento del sistema.
- Problemas en el intercambio de datos.
- Es difícil asegurar la corrección global del sistema
- No hay garantías respecto a qué va a pasar frente a un evento (quién responderá ni en qué orden se dará la ejecución)
- Limitaciones en la verificación (comprobar correctitud debido a dependencia del contexto y secuencia de eventos)

El mecanismo de invocación implícita requiere que el aspecto especifique donde o cuando debe ser invocado. La implementación de un aspecto consecuentemente consiste de dos partes conceptuales diferentes: el código funcional del aspecto (denominado advice) y el código de aplicación del aspecto (denominado pointcut). La funcionalidad del aspecto no es esencialmente diferente del código "regular" y es ejecutado cuando el aspecto es invocado. Esta invocación del aspecto es determinada por el código de aplicabilidad del aspecto.

Hay otros tres tipos de invocación:

- **Excepciones.** Determinadas condiciones provocan la ejecución de un manejador de excepciones asociado, que cambia rápidamente el modo de procesamiento del robot para hacer frente a eventos espontáneos.
- **Wiretapping.** Los mensajes pueden ser interceptados por rutinas que se superponen a la arquitectura existente.
- **Monitores.** Leen información y ejecutan alguna acción si los datos cumplen un cierto criterio. Se dedican a supervisar el sistema