

Diseño de interacción

IES FRANCISCO DE QUEVEDO

HECHO POR: ALAN MIKOLAJCZYK 2DAM

Indice

Introducción	2
Diagrama de secuencia	3
Diagrama de colaboración	4
Conclusión	5

Introducción

Haré un breve resumen sobre que es un diagrama de interacción. Es un diagrama para capturar y describir tanto la estructura estática como el comportamiento dinámico de un sistema. Los diagramas de estado y los de interacción describen el comportamiento para ver como los objetos interactúan dinámicamente en distintos momentos durante la ejecución de un sistema.

Los objetos se comunican enviándose entre ellos mensajes, siendo este una operación donde un objeto llama a otro. En resumen, como los objetos dentro del sistema cambian de estado durante el ciclo de vida y como colaboran a través de la comunicación. Esta comunicación entre objetos se representa con diagramas de interacción y hay 2 tipos: Secuencia y colaboración.

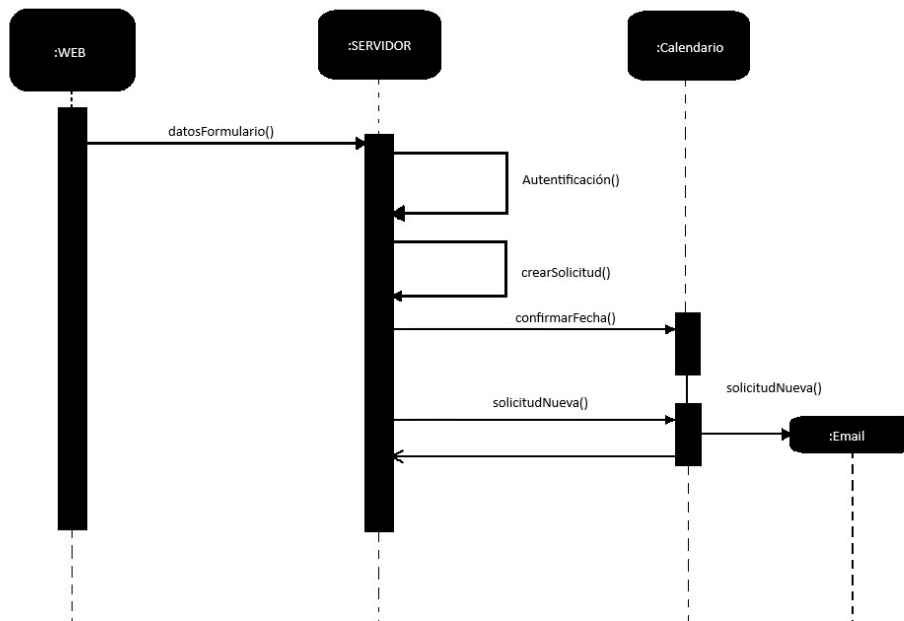
Hablemos de los diagramas de secuencia. Estos nos muestran la interacción de un conjunto de objetos a través del tiempo, o sea, da detalle de los casos de uso (nos proporciona la interacción entre los objetos, que ocurre en el tiempo, para un escenario específico durante la ejecución del sistema).

Finalmente hablaremos de los diagramas de colaboración. Muestran la interacción de un conjunto de objetos en el espacio, o sea, se centra en la organización estructural de los objetos que envían y reciben mensajes.

Habiendo hecho esta breve introducción, a continuación, haré ejemplos tanto del diagrama de secuencia como de colaboración de un mismo sistema con una breve explicación de cómo funciona.

Diagrama de secuencia

Para este caso he creado un diagrama de solicitud de pago de una página web.



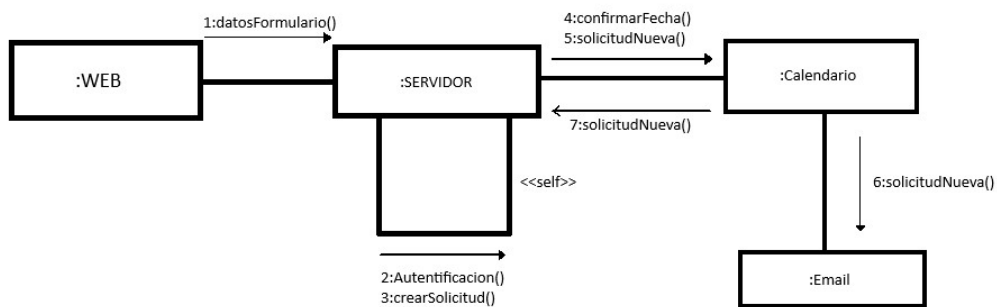
Podemos observar los objetos en los cuadros superiores con ":" delante del nombre del objeto. Las flechas son las llamadas que hay 3 tipos, pero en este caso usamos solo 2, una simple y síncronas. Las simples sirven para transmitir mensajes sin detalles relevantes o que no los hay. Mientras que las síncronas incluye otros mensajes anidados.

Luego aparecen los mensajes que son los textos que van seguidos de las flechas. Estas como podemos ver tienen nombre y pueden contener parámetros, pero en este caso no lo hacen.

Habiendo entendido lo anterior vamos a leerlo. Desde una clase web envías un mensaje a servidor, siendo este "datosFormulario ()". El servidor se auto envía un mensaje "autenticación ()" y "crear solicitud ()". Posteriormente envía un mensaje "confirmarFecha ()" a Calendario. Finalmente, este envía "solicitudNueva ()" a Calendario y este envía esto a Email y a la vez le devuelve un mensaje a Servidor.

Diagrama de colaboración

En este voy a mantener el mismo caso para ver la diferencia.



Como podemos observar los objetos se muestran de la misma manera, pero podemos observar una cosa distinta que ahora hay números delante. Estos son los números de secuencia y estos van en orden. También podemos observar que hay un Path e indican que tipo de objeto recibe el mensaje. Estos van entre "<<>>". Hay varios, por ejemplo: <<local>>, <<global>>, <<parameter>>, <<self>> (aparece en el diagrama).

Las flechas son iguales todas síncronas menos 1 que es simple. Van en orden como antes, pero esta vez tienen número de secuencia, pero ahora la novedad es el path y es el <<self>> que hace referencia a que se envía el mensaje a sí mismo.

Finalmente habría que mencionar la creación y destrucción de un objeto que se muestran entre "<<>>" y hay 3 tipos: new, destroy y transient. Las 2 primeras son obvias y no creo que haga falta explicar, pero transient indica que se crea un objeto durante la ejecución de la interacción, pero se destruye antes de que termine la misma.

Conclusión

Este trabajo me ha traído una comprensión sobre los beneficios de hacer estos diagramas (aunque no se si los he hecho bien) que seguramente necesiten un par de retoques de mejora por mi falta de experiencia ante estos al ser mi primera vez haciendo los.

Podemos sacar como ventajas la organización de las estructuras de diversos eventos interactivos, modelar un sistema como secuencia de eventos ordenados por tiempo, transmitir de manera simple el comportamiento de los mensajes y de las líneas de vida dentro de un sistema o incluso identificar las conexiones entre los elementos de la línea de vida.

Unas desventajas que podría sacar de esto es que el tiempo que se le da a cada mensaje no es el mismo que al de tiempo real de ejecución y aparte no muestra las relaciones que hay entre los objetos.

Podría resumir esto como una experiencia positiva debido a que he aprendido a leer diagramas de interacción que me pueden servir para proyectos futuros mas grandes con un equipo que se dedique a esto sacando todas las ventajas de trabajar con esta herramienta.