|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | **Diagrama de Secuencia UML 2** | http://www.sparxsystems.com/images/layout/h-corner2.gif | |
| **Diagrama de Secuencia**  Un diagrama de secuencia es una forma de **diagrama de interacción** que muestra los objetos como **líneas de vida** a lo largo de la página y con sus interacciones en el tiempo representadas como **mensajes** **dibujados como flechas** **desde la línea de vida origen hasta la línea de vida destino**.  Los diagramas de secuencia son buenos para mostrar qué objetos se comunican con qué otros objetos y qué mensajes disparan esas comunicaciones. Los diagramas de secuencia no están pensados para mostrar lógicas de procedimientos complejos.  **Línea de Vida**  Una línea de vida representa un participante individual en un diagrama de secuencia. Una línea de vida usualmente tiene un rectángulo que contiene el nombre del objeto. Si el nombre es self entonces eso indica que la línea de vida representa el clasificador que posee el diagrama de secuencia.  **http://www.sparxsystems.com/images/screenshots/uml2_tutorial/seq01.GIF**  Algunas veces un diagrama de secuencia tendrá una línea de vida con un símbolo del elemento actor en la parte superior. Este usualmente sería el caso si un diagrama de secuencia es contenido por un caso de uso. Los elementos entidad, control y límite de los diagramas de robustez también pueden contener líneas de vida.  **http://www.sparxsystems.com/images/screenshots/uml2_tutorial/seq02.GIF**  **Mensajes**  Los mensajes se muestran como flechas. Los mensajes pueden ser completos, perdidos o encontrados; síncronos o asíncronos: llamadas o señales. En el siguiente diagrama, el primer mensaje es un mensaje síncrono (denotado por una punta de flecha oscura), completo con un mensaje de retorno implícito; el segundo mensaje es asíncrono (denotado por una punta de flecha en línea) y el tercero es un mensaje de retorno asíncrono (denotado por una línea punteada).  **http://www.sparxsystems.com/images/screenshots/uml2_tutorial/seq03.GIF**  **Ocurrencia de ejecución**  Un rectángulo fino a lo largo de la línea de vida denota la ocurrencia de ejecución o **activación de un foco de control.** En el diagrama anterior hay tres ocurrencias de ejecución. El primero es el objeto origen que envía dos mensajes y recibe dos respuestas, el segundo es el objeto destino que recibe un mensaje asíncrono y retorna una respuesta, y el tercero es el objeto destino que recibe un mensaje asíncrono y retorna una respuesta.  **Mensaje Self**  Un mensaje self puede representar una llamada recursiva de una operación, o un método llamando a otro método perteneciente al mismo objeto. Este se muestra como cuando crea un foco de control anidado en la ocurrencia de ejecución de la línea de vida.  http://www.sparxsystems.com/images/screenshots/uml2_tutorial/seq04.GIF  **Mensajes perdidos y encontrados**  Los mensajes perdidos son aquellos que han sido enviados pero que no han llegado al destino esperado, o que han llegado a un destino que no se muestra en el diagrama actual. Los mensajes encontrados son aquellos que llegan de un remitente no conocido, o de un remitente no conocido en el diagrama actual. Ellos se denotan yendo o llegando desde un elemento de punto final.  http://www.sparxsystems.com/images/screenshots/uml2_tutorial/seq10.GIF  **Inicio y final de línea de vida**  Una línea de vida se puede crear o destruir durante la escala de tiempo representada por un diagrama de secuencia. En el último caso, la línea de vida se termina por un símbolo de detención, representado como una cruz. En el primer caso, el símbolo al inicio de la línea de vida se muestra en un nivel más bajo de la página que el símbolo del objeto que causó la creación. El siguiente diagrama muestra un objeto que fue creado y destruido.  http://www.sparxsystems.com/images/screenshots/uml2_tutorial/seq05.GIF  **Restricciones de tiempo y duración**  En forma predeterminada, un mensaje se muestra como una línea horizontal. Ya que la línea de vida representa el pasaje de tiempo hacia abajo, cuando se modela un sistema en tiempo real, o incluso un proceso de negocios en tiempo límite, puede ser importante considerar el tiempo que toma realizar las acciones. Al configurar una restricción de duración para un mensaje, el mensaje se mostrará como una línea inclinada.  http://www.sparxsystems.com/images/screenshots/uml2_tutorial/seq06.GIF  **Fragmentos combinados**  Se estableció anteriormente que no se espera que los diagramas de secuencia muestren lógicas de procedimientos complejos. Siendo este el caso, hay un número de mecanismos que permiten agregar un grado de lógicas de procedimientos a los diagramas y que a la vez vienen bajo el encabezado de fragmentos combinados. Un fragmento combinado es una o más secuencias de procesos incluidas en un marco y ejecutadas bajo circunstancias nombradas específicas. Los fragmentos disponibles son:   * El fragmento Alternative (denotedo “alt”) modela estructuras if…then…else. * El fragmento Option (denotado “opt”) modela estructuras switch. * El fragmento Break modela una secuencia alternativa de eventos que se procesa en lugar de todo del resto del diagrama. * El fragmento Parallel (denotado “par”) modela procesos concurrentes. * El fragmento de secuenciado Weak (denotado “seq”) incluye un número de secuencias para las cuales todos los mensajes se deben procesar en un segmento anterior, antes de que el siguiente segmento pueda comenzar, pero que no impone ningún secuenciado en los mensajes que no comparten una línea de vida. * El fragmento de secuenciado Strict (denotado “strict”) incluye una serie de mensajes que se deben procesar en el orden proporcionado. * El fragmento Negative (denotado “neg”) incluye una serie de mensajes inválidos. * El fragmento Critical incluye una sección crítica. * El fragmento Ignore declara un mensaje o mensajes que no son de ningún interés si este aparece en el contexto actual. * El fragmento Consider es el opuesto del fragmento Ignore: cualquier mensaje que no se incluya en el fragmento Consider se debería ignorar. * El fragmento Assertion (denotado “assert”) designa que cualquier secuencia que no se muestra como un operando de la aserción es inválida. * El fragmento Loop incluye una serie de mensajes que están repetidos.   El siguiente diagrama muestra un fragmento loop.  http://www.sparxsystems.com/images/screenshots/uml2_tutorial/seq07.GIF  También hay una ocurrencia de interacción, que es similar a un fragmento combinado. Una ocurrencia de interacción es una referencia a otro diagrama que tiene la palabra “ref” en la esquina izquierda arriba del marco, y tiene el nombre del diagrama referenciado que se muestra en el medio del marco  **Puerta** Una puerta es un punto de conexión para conectar un mensaje dentro de un fragmento con un mensaje fuera del fragmento. EA muestra una puerta como un cuadro pequeño en un marco del fragmento.  http://www.sparxsystems.com/images/screenshots/uml2_tutorial/seq08.GIF  **Descomposición en parte**  Un objeto puede tener más de una línea de vida que viene de ésta. Esto permite mensajes de entre e intra objetos para que se muestren en el mismo diagrama.  http://www.sparxsystems.com/images/screenshots/uml2_tutorial/seq11.GIF  **Continuaciones / Invariantes de Estado**  Una invariante de estado es una restricción ubicada en una línea de vida que debe ser verdadera en el tiempo de ejecución. Esta se muestra como un rectángulo con los extremos en semi-circulos.  http://www.sparxsystems.com/images/screenshots/uml2_tutorial/seq09.GIF  Una continuación tiene la misma notación que una invariante de estado pero se usa en fragmentos combinados y puede extenderse a través de más de una línea de vida. |