

REPRESENTACIÓN DE EVENTOS DISPARADORES Y DE RESULTADO EN EL GRAFO DE INTERACCIÓN DE EVENTOS

**Carlos Mario Zapata
Jaramillo**

PhD Ingeniería de Sistemas
Universidad Nacional de Colombia
cmzapata@unal.edu.co

**Paola Andrea Noreña
Cardona**

Ingeniera de Sistemas
IU Tecnológico de Antioquia
panorena@tdea.edu.co

**Nancy Estella González
Granados**

Especialista Ingeniería de Sistemas
Hospital Pablo Tobón Uribe
correongonzalez@hptu.org.co

(Tipo de Artículo: **Investigación**. Recibido el 13/06/2013. Aprobado el 10/12/2013)

RESUMEN

El Grafo de Interacción de Eventos (EIG por sus siglas en inglés) es una herramienta que representa la relación y ocurrencia de los eventos en una aplicación de software. Algunos autores utilizan este grafo en la lógica de negocio para representar la interacción de eventos en el sistema, generalmente, en operaciones y funciones de la interfaz gráfica de usuario (GUI por sus siglas en inglés). Los eventos disparadores y de resultado son responsables de los cambios en el estado de los procesos de negocio. Sin embargo, no existen artefactos que representen la interacción entre ellos. Es por esta razón que en este artículo se propone una adaptación del EIG para la representación y la interacción de los eventos disparadores y de resultado en el proceso de ingeniería de requisitos. Se emplea como base para la representación la notación de los eventos en esquemas preconceptuales.

Palabras clave

Evento, evento de resultado, evento disparador, grafo de interacción de eventos, ingeniería de requisitos.

REPRESENTING TRIGGER AND RESULT EVENTS IN THE EVENT INTERACTION GRAPH

ABSTRACT

The Event Interaction Graph (EIG) is a tool which represents the relation and occurrence of events in a software application. Some authors commonly use this graph in business logic for representing the event interaction within the system. In this context, events are linked to operations and functions of the graphical user interface (GUI). Trigger and result events are responsible for changing the state of the business processes. However, there are no artifacts representing the interaction among these events. For this reason, in this paper we propose an adaptation of the EIG for representing the interaction of trigger and result events in requirement engineering processes. We employ a pre-conceptual-schema-based notation as a basis for representing the events.

Keywords

Event, Event interaction graph, Requirement engineering, Result event, Trigger event.

REPRÉSENTATION D'ÉVÉNEMENTS DÉCLENCHEURS ET DE RÉSULTATS DANS LE GRAPHE D'INTERACTION D'ÉVÉNEMENTS

RÉSUMÉ

Le Graphe d'Interaction d'Événements (EIG, selon son sigle anglais) est un outil qui représente la relation et occurrence des événements dans une application logicielle. Quelques auteurs utilisent ce graphe dans la logique des affaires pour représenter l'interaction d'événements dans le système, principalement dans les opérations et fonctions de l'interface graphique d'utilisateur (GUI, selon son sigle anglais). Les événements déclencheurs et de résultats sont responsables des changements d'état des processus des affaires. Cependant il n'y a pas des objets qui représentent l'interaction entre eux. Par conséquent dans cet article on propose une adaptation de l'EIG pour la représentation et l'interaction des événements déclencheurs et de résultat dans le processus de l'ingénierie des exigences. Nous utilisons comme référence pour la représentation la notation des événements dans schémas pre-conceptuels.

Mots-clés

Événement, événement de résultat, événement déclencheur, graphe d'interaction d'événements, ingénierie des exigences.

1. INTRODUCCIÓN

Los eventos en la interfaz gráfica de usuario (GUI, por sus siglas en inglés¹) se asocian con operaciones y funciones que se pueden realizar con el *mouse*. Existen modelos de representación de estos eventos que se denominan grafos de eventos y se utilizan como herramientas que contienen nodos y aristas para simbolizar los sucesos que ocurren en un sistema [6] como: el grafo de flujo de eventos (EFG) [7], [8], el grafo de interacción semántica de eventos (ESIG) [9], el grafo de secuencias (ESG) [10] y el grafo de interacción de eventos (EIG) [11], los cuales representan posibles interacciones entre eventos que se pueden ejecutar en la GUI. A diferencia de los demás grafos, el EIG es una herramienta automática que representa la relación y ocurrencia de los eventos en una aplicación de software. Algunos autores [9], [11], [12], [18] utilizan este último en la lógica de negocio para representar la interacción de los eventos en la GUI. Adicionalmente, este grafo es un refinamiento del EFG [12].

Los eventos en una aplicación de software son sucesos que ocurren en un momento determinado, proporcionan información relevante sobre el funcionamiento del sistema y son importantes en la definición de los requisitos funcionales. Los eventos son responsables de habilitar la ocurrencia de cambios de estado en los procesos de negocio como la finalización y el inicio de dichos procesos [1]. Cuando un evento inicia un proceso se denomina disparador y cuando lo finaliza se denomina de resultado [2], [3].

La ingeniería de requisitos implica la comprensión de las necesidades y expectativas del interesado durante el proceso de educación de requisitos [13], [14]. Los eventos disparadores y de resultado son importantes en la ingeniería de requisitos porque permiten definir requisitos funcionales y dan información relevante del funcionamiento del sistema [1]. De acuerdo con esto, se requiere un artefacto que permita representar la interacción de los eventos disparadores y de resultado para una mejor solución de las necesidades del interesado.

Por consiguiente, en este artículo se propone una adaptación del EIG para su representación en el proceso de ingeniería de requisitos, tomando como base la notación de los eventos en esquemas preconceptuales (PS) ya que estos esquemas presentan la notación de eventos de forma clara y específica y, también, se utilizan para representar de forma controlada el dominio del discurso de los

interesados, de tal manera que cualquier tipo de persona los pueda interpretar [4].

La estructura del artículo es la siguiente: en la Sección 2 se presenta una breve descripción de los eventos, la especificación de los tipos de eventos y la notación de los PS para representar eventos; en la Sección 3 se exponen los grafos de eventos en la interfaz gráfica de usuario; en la Sección 4 se justifica el porqué de la necesidad de representar eventos disparadores y de resultado en el proceso de ingeniería de requisitos; en la Sección 5 se presenta la propuesta para la representación de eventos disparadores y de resultado mediante el grafo de interacción de eventos y, finalmente, en la Sección 6 se presentan las conclusiones y el trabajo futuro.

2. EVENTOS

En ingeniería de software, los eventos habilitan la ocurrencia de cambios de estado en sus procesos, como por ejemplo la finalización y el inicio de los mismos. Los eventos proporcionan información relevante sobre el funcionamiento del sistema y son importantes en la definición de sus requisitos funcionales [1].

Un evento es algo que sucede durante el curso de un proceso de negocio. Estos eventos afectan el flujo del proceso y suelen tener una causa mediante un disparo o un impacto como resultado [15]. El término *evento* es lo suficientemente general como para abarcar muchas cosas en un proceso de negocio. El inicio o el final de una actividad, el cambio de estado de un proceso, un mensaje que llega, el fin de un proceso, etc., se pueden considerar como eventos. Sin embargo, la notación para el modelado de procesos de negocio (*Business Process Modeling Notation* BPMN) restringe el uso de eventos para incluir sólo los tipos de eventos que afectarán la secuencia de tiempo de las actividades de un proceso.

Los eventos se representan con círculos con centros abiertos que permiten incluir marcadores internos para distinguir diferentes factores desencadenantes o de resultados. Hay dos tipos de eventos, que afectan al flujo: disparadores y de resultado [3].

2.1. Evento disparador y de resultado

Cuando un evento inicia un proceso se denomina evento disparador y cuando lo finaliza se denomina evento de resultado [2]. En el modelado de procesos de negocio, un evento sucede por la ejecución de un proceso; el orden de ocurrencia se da por un disparo y

¹ Con el fin de conservar el estándar internacional de las siglas de este artículo, todas las siglas se manejarán en inglés.

como consecuencia de un proceso se obtiene un evento de resultado. El evento disparador indica el punto de partida del proceso y el evento de resultado el fin del mismo. Los eventos se disparan, generalmente, cuando el último paso del proceso anterior se completa [16].

De acuerdo con lo que se especifica en BPMN, un evento disparador inicia el flujo o la secuencia de un proceso, es decir, solo tiene flujos de secuencia de salida. También, puede desencadenar la ejecución de otros eventos [17]. Se representa gráficamente con un círculo de línea delgada [16] como lo muestra la Figura 1.

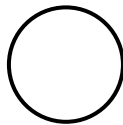


Fig. 1. Notación del evento disparador en BPMN.

En BPMN hay seis (6) tipos de eventos disparadores, cuatro (4) básicos: simple, mensaje, temporizador y señal; y dos (2) de tipo avanzado: condicional y múltiple, que se describen en la tabla 1 [16].

De igual manera BPMN define el evento de resultado, el cual indica el final de un proceso, en términos de flujo de secuencia que muestra el fin del flujo de un proceso, por ejemplo un mensaje se envía o una señal se transmite [19]. Gráficamente, se simboliza con un círculo de línea gruesa para que se distingan de los eventos de inicio como se puede observar en la figura 2. Al igual que en el caso anterior se pueden colocar marcadores dentro del círculo para mostrar variaciones del evento [16].

Al igual que en los eventos disparadores, se agrupan según su tipo en cuatro (4) básicos así: simple, mensaje, señal y terminador y cuatro (4) avanzados: error, cancelación, compensación y múltiple. En la tabla 2 se presenta la definición de cada uno [16].

2.2. Diagramas que representan eventos disparador y de resultado de procesos

2.2.1. Diagrama de procesos de negocio BPMN

Muestran como, durante el modelado de procesos de negocio, la representación de los eventos afecta los flujos de procesos. Un evento inicia un flujo de procesos, ocurre durante este flujo o lo termina. BPMN proporciona una notación distinta para cada uno de estos tipos de eventos.

Tabla 1. Tipos de eventos disparadores en BPMN

Evento	Descripción	Símbolo
Simple	No define una condición o requisito para iniciar un proceso o subproceso.	
Mensaje	El disparador es un mensaje que llega desde otra entidad o rol (participante)	
Temporizador	Especifica una fecha-hora para activar el inicio de un proceso.	
Condicional	Es un disparador con una expresión de condición que se debe satisfacer para que empiece el proceso.	
Señal	Puede llegar desde otro proceso y dispara el inicio del proceso. Una señal no es un mensaje. Varios procesos pueden tener eventos de arranque que se desencadenan a partir de la misma señal emitida.	
Múltiple	Hay uno o más disparadores (mensajes, condiciones, señales o temporizadores) que pueden iniciar un proceso.	

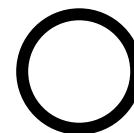










Fig. 2. Notación de evento de resultado en BPMN.

A menudo ocurre un evento mientras que un proceso en particular se está realizando, causa una interrupción en el proceso o provoca un nuevo proceso a realizar o se completa un proceso, que causa el inicio de un nuevo evento.

BPMN cuenta con una representación definida para los eventos que incluye mensajes, temporizadores, reglas de negocio y condiciones de error. Además permite especificar el tipo de disparo del evento y lo denota con un icono representativo, tal como se especifica en la figura 2. El patrón que se observa en el diagrama de procesos es que un evento se representa con un círculo. La definición de BPMN se puede observar en las tablas 1 y 2 y las figuras 1 y 2.

Tabla 2. Tipos de eventos de resultado en BPMN.

Evento	Descripción	Símbolo
Simple	No se define ningún resultado.	
Mensaje	Comunicación con otra entidad de negocio. (Participante o proceso).	
Señal	Define un evento "broadcast" el cual cualquier otro proceso puede ver y al cual puede reaccionar.	
Terminador	Suspende todas las actividades de un proceso.	
Error	Un estado final que interrumpe el proceso o que requiere corrección.	
Cancelación	Indica que el fin del camino de un proceso resulta de la cancelación de un sub-proceso de transacción.	
Compensación	Indica que el final del camino de un proceso resulta en la activación de una compensación.	
Múltiple	Representa dos o más eventos de resultado.	

2.2.2. Esquemas preconceptuales

El proceso de educación de requisitos en ingeniería de software es punto crítico para el éxito de un proyecto de diseño e implementación de un sistema. Es por esto que el analista debe emplear artefactos que logren representar de manera adecuada el problema que el interesado quiere solucionar y para ello se pueden utilizar los esquemas preconceptuales (PS). Estos esquemas permiten diagramar de manera controlada conceptos, relaciones y restricciones del discurso del dominio del interesado. Además de ser de fácil lectura para cualquier tipo de persona, también es posible construir a partir de ellos diferentes PSs. Los elementos utilizados y la descripción que se utiliza para su construcción se muestran a continuación [4]:

Conceptos: son sustantivos o sintagmas nominales obtenidos del discurso del interesado.

Relaciones estructurales: son relaciones permanentes entre los conceptos y se asocian con los verbos "es" y "tiene".

Relaciones dinámicas: generan relaciones de tipo temporal con los conceptos y se asocian con los denominados "verbos de actividad".

Relaciones eventuales: Pueden ser verbos en presente simple para los eventos disparadores y verbos en participio pasado para eventos de resultado [5].

Condicionales: indican las restricciones o reglas del negocio que se deben cumplir.

Implicaciones: se usan para establecer una relación causa-efecto entre las relaciones dinámicas o también unen condicionales con relaciones dinámicas.

Conexiones: permiten enlazar conceptos con relaciones y viceversa.

Enlaces de operador: permiten conectar operadores en condiciones.

Aglutinadores: Son aquellos que contienen otros elementos. Pueden ser especificación, restricción, marco y eventual.

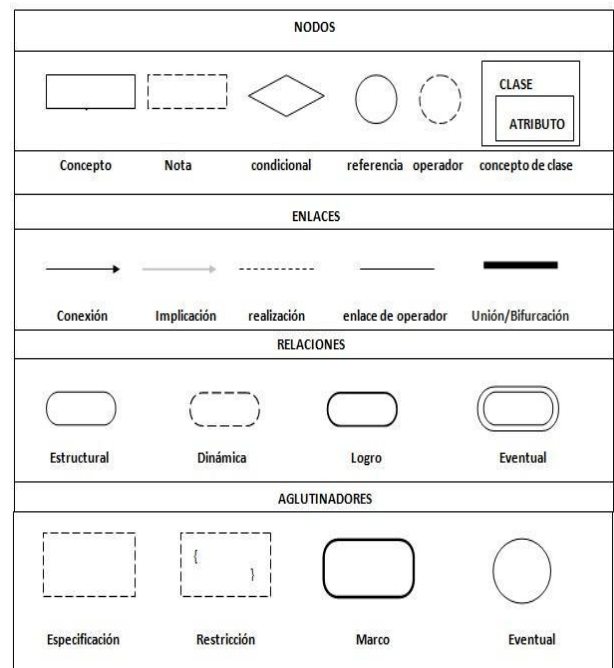


Fig. 3. Elementos de los PSs.

En la notación de los PSs se toma el círculo de la representación en BPMN, una relación eventual que determina el tipo de evento que se relaciona con el nombre del concepto mediante una flecha de conexión. Los eventos así definidos se unen con implicaciones a las relaciones dinámicas. La relación eventual se define como un verbo en participio pasado o un verbo

intransitivo y se representa con un óvalo en doble línea [5]. Si el verbo es intransitivo y se conjuga en presente simple, el evento es disparador y la implicación entra a la relación dinámica. Si el verbo se maneja en participio pasado, se trata de un evento de resultado y la implicación sale de la relación dinámica, como se muestra en la figura 4.

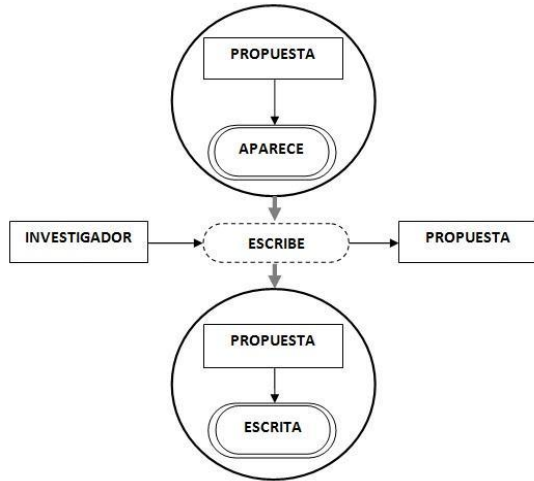


Fig. 4. Notación del evento disparador y de resultado en PSs.

3. GRAFO DE EVENTOS

El grafo de eventos utiliza nodos y aristas para simbolizar los eventos que ocurren en un sistema. Los nodos representan declaraciones o expresiones de cálculo y las aristas la transferencia de control entre los nodos [6]. También, representan de manera gráfica los modelos de tiempo discreto, en donde los nodos que simbolizan los eventos se conectan por medio de arcos dirigidos (flechas o aristas) los cuales muestran cómo los eventos se planifican a partir de otros o de ellos mismos. Las aristas se asocian con una condición que pueden ser falsa o verdadera, además de un tiempo t que simboliza un retardo para la ejecución del evento [20], [21].

La figura 5 muestra la estructura básica de un grafo y se puede interpretar así: cuando se ejecuta el evento A se evalúa la condición (i) y, si es verdadera, entonces el evento B se programa después de un retraso de tiempo t .

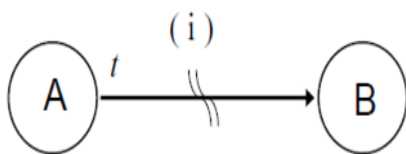


Fig. 5. Estructura básica de los grafos de eventos.

Para la representación del grafo se utiliza de manera estándar la ubicación de la simbología, de tal manera

que el retraso de tiempo t se ubica en el extremo donde inicia la arista, la condición o procesos (i) en la parte central y por encima de las líneas serpenteantes que la cortan. Cuando en una transición no exista un retraso, éste se omite en la representación, y si el evento B ocurre siempre que se ejecuta el evento A, entonces también se omite la representación de la condición en el grafo y, en este caso, se dice que la arista es incondicional. Existen diferentes tipos de grafos que se utilizan para representar los eventos específicamente en la GUI, que se definen seguidamente.

3.1. Grafo de flujo de eventos (EFG):

Es un método de modelado de la GUI para crear casos de prueba que representan las posibles secuencias de eventos que se pueden ejecutar en un sistema [7], [8], como se muestra en la figura 6.

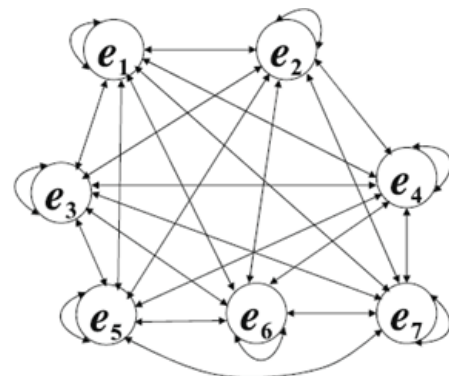


Fig. 6. Grafo de Flujo de Eventos (EFG).

3.2. Grafo de interacción semántica de eventos (ESIG):

Utiliza los estados de la GUI como retroalimentación para identificar las aristas importantes. Además, genera nuevos casos de prueba más largos que cubren las aristas [9].

3.3. Grafo de secuencia de eventos (ESG):

Se utiliza para graficar un subconjunto de interacciones entre un sistema y su usuario (véase la figura 7). El conjunto completo de interacciones se captura en términos de un conjunto de ESG, donde cada ESG representa un conjunto posiblemente infinito de secuencias de eventos basadas en un EIG [10].

3.4. Grafo de interacción de eventos (EIG):

Son herramientas automáticas que representan la relación y ocurrencia de los eventos de la GUI para generar casos de prueba entre los eventos. Las ocurrencias son sincronizaciones de las unidades del programa en cualquier instante y las relaciones son interacciones entre dos o más nodos en una secuencia de eventos [18]. Los EIG en la lógica de negocio se usan para representar la relación y ocurrencia entre los

eventos y utilizan como elementos principales nodos y aristas. Los EIG también representan todas las posibles secuencias de eventos que se pueden ejecutar en la interfaz gráfica de usuario y la interacción entre eventos y procesos [11]. La abstracción del EIG se basa en los eventos que interactúan en una aplicación [9], a diferencia de los demás modelos que representan los eventos de naturaleza estructural de la GUI como abrir o cerrar ventanas. Adicionalmente, este grafo es un refinamiento del EFG [12] y a partir de este modelo se define el ESIG. Su imagen se aprecia en la figura 8.

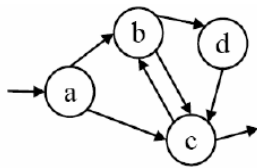


Fig. 7. Grafo de Secuencia de Eventos (ESG).

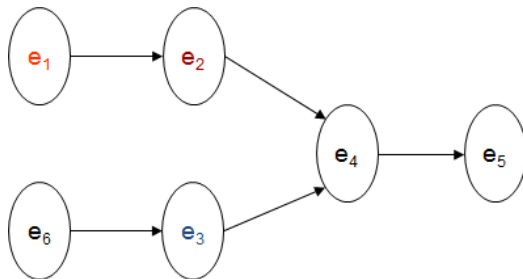


Fig. 8. Grafo de Interacción de Eventos (EIG).

4. NECESIDAD DE REPRESENTAR LA INTERACCIÓN DE LOS EVENTOS DISPARADORES Y DE RESULTADO

La ingeniería de requisitos implica la comprensión de las necesidades y expectativas del interesado durante el proceso de educación de requisitos [13], [14]. Los eventos disparadores y de resultado son importantes en la ingeniería de requisitos porque permiten definir requisitos funcionales y dan información relevante del funcionamiento del sistema [1]. Además, es necesario destacar que se encargan de cambiar el estado de los procesos lo que afecta el flujo de la secuencia de las actividades [15]. En el proceso de ingeniería de requisitos existen varios artefactos que representan eventos, entre ellos: el diagrama de procesos de negocio, el diagrama de máquina de estados y el PS, entre otros. Sin embargo, no se encuentran artefactos que permitan representar la interacción entre los eventos disparadores y de resultado.

En la interfaz gráfica de usuario se utilizan diferentes modelos de representación de eventos para la creación de casos de prueba. Estos eventos se asocian exclusivamente con las funciones y operaciones de la

GUI; entre esos modelos están: EFG, ESG, ESIG y EIG. Algunos autores [9], [11], [12], [18] utilizan este último para representar la interacción de los eventos en la GUI, aunque tampoco para representar la interacción de los eventos disparadores y de resultado.

De acuerdo con lo anterior, se presenta la necesidad de un artefacto que permita representar la interacción (relación, el orden de ocurrencia y la secuencia) entre los eventos disparadores y de resultado en el proceso de ingeniería de requisitos. Tal artefacto permitiría a los analistas una representación completa del sistema y una mejor implementación de la solución a los problemas de un interesado. Se plantea una adaptación del EIG que permita visualizar la relación, el orden de ocurrencia y la secuencia entre los eventos. Se elige la notación en PSs que proporciona una representación específica de los eventos disparadores y de resultado, de tal manera que cualquier tipo de persona lo pueda interpretar [4].

5. REPRESENTACIÓN DE EVENTOS DISPARADORES Y DE RESULTADO EN EL GRAFO DE INTERACCIÓN DE EVENTOS

En esta sección se pretende realizar una adaptación del grafo de interacción de eventos que permita la representación de la interacción de eventos disparadores y de resultado, empleando la notación de los PSs.

5.1. Símbolos para representar eventos

De la notación de eventos en PSs se pretende utilizar los siguientes elementos: operador, condicional, especificación, concepto, conexión, evento y relación eventual.

Se le añadirá al círculo que representa los eventos un color diferente para ambos tipos de eventos. Para los eventos de resultado el color azul y el color rojo para los eventos disparadores, tal como se evidencia en la figura 9.



Fig. 9. Símbolos representación de Eventos

5.2. Representación de los eventos de resultado

De acuerdo con la notación anterior un evento de resultado se compone de un concepto, una flecha de

conexión y una relación eventual que contiene un verbo en participio pasado. Dentro del EIG se puede unir un evento de resultado a un evento disparador significando que es el orden en la secuencia más no que proviene del evento de resultado. Dentro de este grafo no es adecuado poner dos eventos de resultado continuos.



Fig. 10. Notación para eventos de resultado.

5.3. Representación de los eventos disparadores

Se pueden representar de dos formas: una en la que se conforma por un concepto, una flecha de conexión y una relación eventual que contiene un verbo en presente simple y la otra en la que se define una condición mediante un condicional y especificaciones relacionadas con conexiones de operador. En el grafo de interacción de eventos, un evento disparador desencadena otro evento disparador o un proceso.

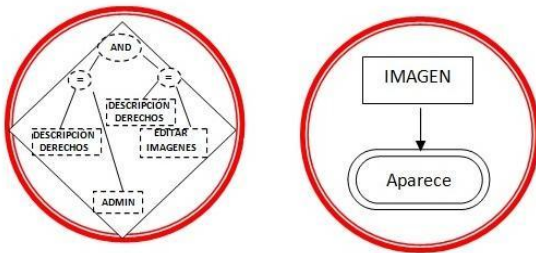


Fig. 11. Notación para eventos disparadores.

5.4. Representación de procesos

En el cambio de estados de los procesos, los eventos inciden para dispararlos o pueden ser un resultado de estos, por lo tanto es importante representarlos en el EIG. Para la representación de estos procesos se utiliza una línea gruesa vertical con su nombre respectivo que precede un proceso o un evento de resultado.

5.5. Estructura completa para la representación de eventos disparadores y de resultado en un grafo de interacción de eventos

Uniendo todas las piezas, se presenta un ejemplo donde el dominio es el proceso en las redes sociales para manipular un álbum, con imágenes y etiquetas. Este grafo de interacción de eventos contiene 11 nodos de los cuales cinco son eventos disparadores y seis son eventos de resultado. También tiene 7 procesos en la interacción entre eventos.

Eventos disparadores

Usuario llega, Condición: si es admin y tiene permisos para manejar todos los álbumes, Imagen aparece, Condición: si es admin y tiene permisos para subir imágenes, Condición: si es admin y tiene permisos para etiquetar.

Eventos de resultado

Álbum publicado, Álbum editado, Álbum eliminado, Imagen subida, Imagen editada, La imagen es etiquetada.

Procesos

Crear álbum, Publicar álbum, Editar álbum, Eliminar álbum, Subir imagen, Editar imagen, Etiquetar imagen

5.5.1. Secuencia de eventos

En este grafo de interacción de eventos se representan todas las posibles secuencias de la relación y ocurrencia entre los eventos los cuales son:

Camino 1:

El usuario llega - si es admin y tiene permisos para manejar todos los álbumes -crea el álbum - publica el álbum el álbum es publicado -aparece una imagen - si es admin y tiene permisos para subir imágenes - sube imagen - la imagen es subida y si es admin y tiene permisos para etiquetar - etiqueta imagen - la imagen es etiquetada.

Camino 2:

El usuario llega - si es admin y tiene permisos para manejar todos los álbumes -crea el álbum - edita el álbum - el álbum es editado - aparece una imagen - si es admin y tiene permisos para subir imágenes - sube imagen - la imagen es subida y si es admin y tiene permisos para etiquetar - etiqueta imagen - la imagen es etiquetada.

Camino 3:

El usuario llega - si es admin y tiene permisos para manejar todos los álbumes -crea el álbum - publica el álbum - el álbum es publicado - aparece una imagen - si es admin y tiene permisos para subir imágenes - Edita imagen - la imagen es editada - si es admin y tiene permisos para etiquetar - etiqueta imagen - la imagen es etiquetada.

Camino 4:

El usuario llega si es admin y tiene permisos para manejar todos los álbumes -crea el álbum edita el álbum el álbum es editado - aparece una imagen si es admin y tiene permisos para subir imágenes edita imagen - la imagen es editada si es admin y tiene

permisos para etiquetar - etiqueta imagen - la imagen es etiquetada.

Camino 5:

El usuario llega si es admin y tiene permisos para manejar todos los álbumes -crea el álbum - elimina el álbum - el álbum es eliminado.

6. CONCLUSIONES Y TRABAJO FUTURO

El campo de los eventos disparador y de resultado es importante explorarlo en la ingeniería del software ya que ellos ejercen un rol significativo dentro de los procesos de negocio. En efecto, los eventos tienen la responsabilidad de generar un control en la ejecución de los procesos de un proyecto de software.

En este artículo se describieron los eventos disparador y de resultado, su representación en PSs y los diferentes modelos que existen para representar eventos en la GUI, explicando la diferencia entre ellos. Igualmente, se propuso un artefacto para la

representación de la interacción de eventos disparador y de resultado en el proceso de ingeniería de requisitos empleando la notación de PSs. Este artefacto permite la representación de la relación, orden de ocurrencia y secuencias de los eventos.

En ingeniería de requisitos, la representación de la interacción de eventos le permite al analista una completa representación la relación, el orden de ocurrencia y las secuencias de los eventos en un proceso de negocio para una correcta implementación y satisfacción de las necesidades del interesado.

Como posibles temas de trabajo futuro que se pueden continuar de este artículo, se mencionan los siguientes:

- La integración del Grafo de interacción de eventos disparador y de resultado en UNC-Method.
- La definición de reglas de consistencia para los eventos disparador y de resultado en algunos artefactos UNC-Method.

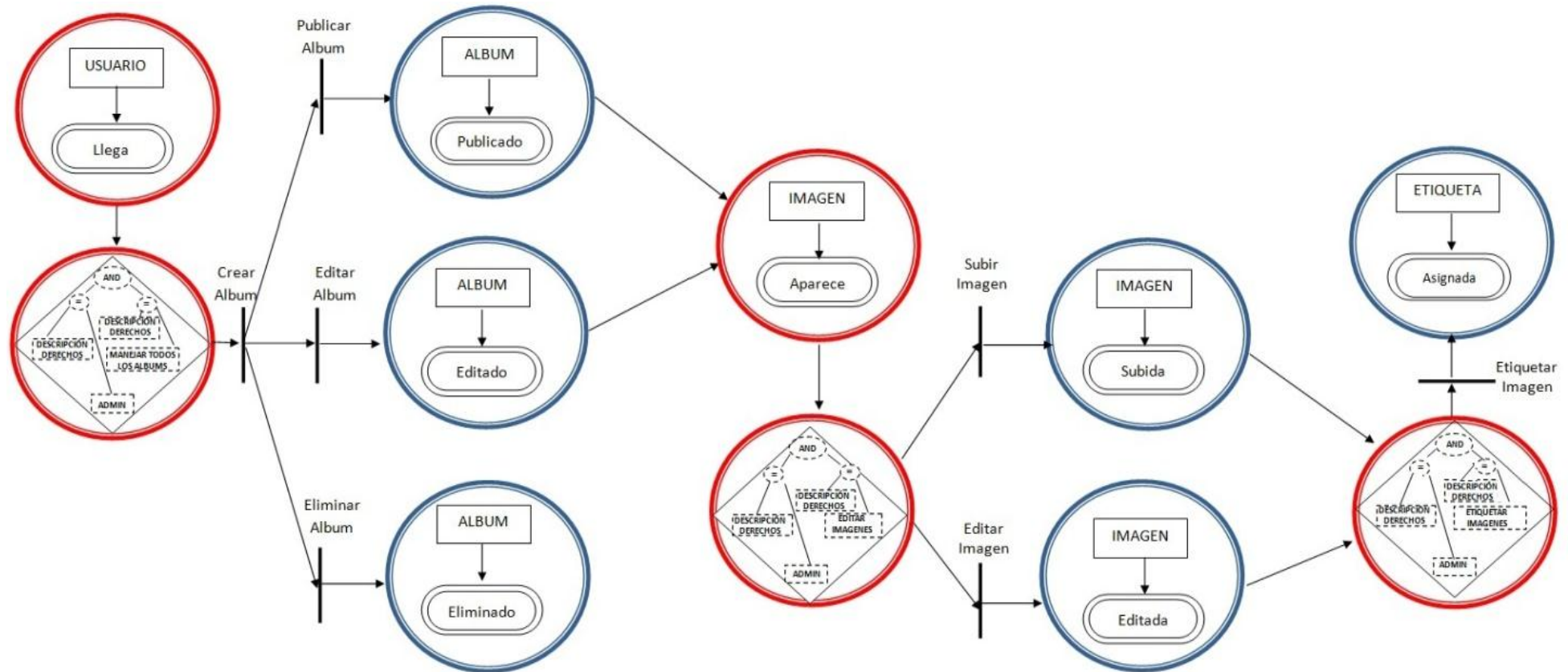


Fig. 12. Adaptación del EIG para eventos disparador y de resultado

REFERENCIAS

- [1] N. Weinbach and A. García. “Una Extensión de la Programación en Lógica que incluye Eventos y Comunicación”. 2004.
- [2] F. Arango and C.M. Zapata. “UN-MÉTODO para la Elicitación de Requisitos de Software”. 2006.
- [3] ORACLE. “Documento número: B14220-02. Documento estándar”. Online [Septiembre, 2012].
- [4] C.M. Zapata; G. Giraldo and S. Londoño. “Esquemas Preconceptuales Ejecutables”. Revista Avances en Sistemas e Informática. 2011.
- [5] C.M. Zapata. “The UNC-Method revisited: elements of the new approach”. Saarbrucken. 2012.
- [6] A.H. Watson and T.J. McCabe. “Structured testing: a testing methodology using the cyclomatic complexity metric”. Technical Report NIST 500-225, 1996.
- [7] A.M. Memon; M.E. Pollack and M.L. Soffa. “Hierarchical GUI Test Case Generation Using Automated Planning”. *IEEE Trans. Software Engineering*, vol. 27, no. 2, pp. 144-155. 2001.
- [8] A.M. Memon. “An event-flow model of GUI-based applications for testing”. *Software testing, verification and reliability*, vol. 17, pp. 137–157. 2007.
- [9] X. Yuan; A. Memon. “Generating Event Sequence-Based Test Cases Using GUI Runtime State Feedback”. *IEEE Trans. Software Engineering*, vol.36, no.1, pp.81-95. 2010.
- [10] F. Belli; N. Nissanke; C.J. Budnik and A. Mathur. “Test Generation Using Event Sequence Graphs”. *Technical Reports and Working Papers*. Sep. 2005.
- [11] Katayama, T.; Furukawa, Z.; Ushijima, K., “Event interactions graph for test-case generations of concurrent programs” *Software Engineering Conference. Proceedings, 1995 Asia Pacific*, vol., no., pp.29-37. Diciembre. 1995.
- [12] X. Yuan; M.B. Cohen and A.M. Memon. “GUI interaction testing: Incorporating event context”. *IEEE Trans. Software Engineering*, vol.99, 2010.
- [13] R. Young “The Requirements Engineering Handbook”. Norwood, MA, USA: Artech House, Incorporated. 2003.
- [14] D.E. Damian and D. Zowghi. “The Impact of Stakeholders Geographical Distribution on Managing Requirements in a Multi-Site Organization”. *Proceedings IEEE Joint International Conference on Requirements Engineering (RE'02)*. Essen, Germany, pp.319-330, 2002.
- [15] U. Hernández; F.J. Alvarez and M. Vargas. “Use Processes - Modeling Requirements Based on Elements of BPMN and UML Use Case Diagrams”. *2nd International Conference on Software Technology and Engineering (ICSTE)*. 2010.
- [16] OMG (Object Management Group). (2009). “Business Process Model and Notation BPMN”. *Standard Document*. <http://www.omg.org/spec/BPMN/1.2>. 2009.
- [17] S. Sabharwal, S. Kumar and D. Sabharwal. “An Event-Based Approach To Generate Test Scenarios”. *Conf. on Computer & Communication Technology*. pp. 551-556. 2010.
- [18] X. Yuan and A. Memon. “Iterative execution-feedback model-directed GUI testing”. *Information and Software Technology*, vol. 52. pp. 559–575. 2010.
- [19] S.A. White; M. Derek. “Guía de Referencia y Modelado BPMN”. ISBN: 978-0-9819870-3-3, 2009.
- [20] A. Urquía M. “Simulación. Texto Base de Teoría”. Departamento de Informática y Automática. Escuela Técnica Superior de Ingeniería Informática, UNED. Madrid, España, 2003, p. 321.
- [21] Y. Pan and Zhou and L. Wang. “Modeling Internet Applications with Event Graphs”. *Sixth International Conference on Fuzzy Systems and Knowledge Discovery*. vol.1, pp. 124-128. 2009.