



UNIVERSIDAD NACIONAL DE COLOMBIA

Un mecanismo de consistencia en los eventos disparador y de resultado para los artefactos de UNC-Method

Paola Andrea Noreña Cardona

Universidad Nacional de Colombia

Facultad de Minas, Departamento de Ciencias de la Computación y de la Decisión

Medellín, Colombia

2013

Un mecanismo de consistencia en los eventos disparador y de resultado para los artefactos de UNC-Method

Paola Andrea Noreña Cardona

Tesis de investigación presentada como requisito para optar al título de:
Magister en Ingeniería de Sistemas

Director:

Ph.D. Carlos Mario Zapata Jaramillo

Línea de Investigación:

Ingeniería del software

Universidad Nacional de Colombia

Facultad de Minas, Departamento de Ciencias de la Computación y de la Decisión

Medellín, Colombia

2013

Dedicatoria

A Dios por darme su amor y su sabiduría, por su respaldo en todo el proceso de mi Maestría.

A mis padres por darme la vida, por su esfuerzo y dedicación en la construcción de mi futuro profesional.

A mi familia por su constante apoyo y compañía

Agradecimientos

A Colciencias, Departamento Administrativo de Ciencia, Tecnología e Innovación en Colombia por su aporte y financiación en el proyecto de investigación.

Al Doctor Carlos Mario Zapata, por su generosa dirección en mi Tesis de Maestría, por sus conocimientos, compromiso y dedicación; por su esfuerzo y contribución en mi formación como Magister en Ingeniería de Sistemas.

Resumen

Los eventos en el ciclo de vida del software generan cambios de estado en los procesos de negocio, afectando el flujo de los procesos. Cuando un evento desencadena un proceso se denomina evento disparador y cuando genera un impacto como resultado de un proceso se denomina evento de resultado. UNC-Method es el método para la educación de requisitos de la Universidad Nacional de Colombia, que utilizan los estudiantes de ingeniería de sistemas para determinar el problema y la solución de un proyecto de software y como parte de su formación. La consistencia en el proceso de ingeniería de software es un aspecto importante ya que permite la representación y articulación de los elementos en sus diferentes fases. En algunos artefactos que representan eventos en UNC-Method, se observan inconsistencias, pues no existe una completa representación de los eventos, y no hay un artefacto que muestre la interacción entre ellos. Además, los eventos se suelen confundir con acciones u operaciones que ejecutan los usuarios.

En esta Tesis de Maestría se define un mecanismo de consistencia para los eventos disparadores y de resultado en artefactos de UNC-Method, que precisa las estructuras gramaticales, las reglas de consistencia y el grafo de interacción para los eventos. Adicionalmente se propone la validación del mecanismo y un prototipo que apoye el mecanismo en las reglas de consistencia.

Palabras claves: Ciclo de vida del desarrollo de software, eventos, evento disparador, evento de resultado, mecanismo de consistencia, UNC-Method.

Abstract

Events generate state changes of the business processes and they affect the flow of the processes in the software development lifecycle. When an event initiates a process is named *trigger event* and when an event generates an impact as the result of a process, it is named *result event*. UNC-Method is a requirements elicitation method the engineering system students use to determinate the problem and the solution of a software project and their academic formation in the Universidad Nacional de Colombia. The consistency is an important aspect in Software Engineering process because it permit the representation and articulation of the elements en its phases. So, we observe inconsistencies in the event representation in some artifacts of UNC-Method: the event representation is incomplete, no artifacts are defined in order to show the interaction among events, and the events are sometimes confused with actions and operations that users execute.

In this M.Sc. Thesis, we define a consistency mechanism for the trigger and result events in artifacts of UNC-Method. Such a mechanism includes some event grammatical structures, the consistency rules and the event interaction graph. Additionally we propose the validation of the mechanism and a software prototype for supporting the consistency rules in the mechanism.

Keywords: Consistency mechanism, event, Result event, Trigger event, software development lifecycle.

Contenido

	Pág.
Abstract	vi
Contenido.....	vii
Lista de Figuras	ix
Lista de Tablas.....	10
1. Alcance y planteamiento del problema de investigación	13
1.1 Antecedentes	13
1.2 Problema.....	17
1.3 Objetivos	18
1.3.1 General	18
1.3.2 Específicos.....	18
1.4 Hipótesis	18
1.5 Motivación.....	19
1.6 Metodología.	20
1.6.1 Problema de investigación	20
1.6.2 Análisis.....	20
1.6.3 Diseño de la solución	20
1.6.4 Validación de la solución.....	21
2. Marco teórico	22
2.1 Marco referencial	22
2.1.1 Eventos	22
2.1.2 Tipos de eventos.....	23
2.1.3 UNC-Method	26
2.1.4 Consistencia.....	28
2.1.5 Mecanismo.....	28
2.2 Dominio del problema	29

2.2.1	Representación de eventos en artefactos de UNC-Method	29
2.2.2	Diálogo controlado	29
2.2.3	Tarjetas de educación de requisitos.....	31
2.2.4	Esquemas preconceptuales	32
2.2.5	Diagrama de procesos	34
2.2.6	Tabla explicativa del diagrama de procesos	35
2.2.7	Diagrama de Máquina de estados	36
3. Solución propuesta: Mecanismo de consistencia.....		38
□	Diagrama de procesos de negocio BPMN	38
□	Diagrama de máquinas de estado UML.....	39
3.1	Estructuras gramaticales de los eventos.....	41
3.1.1	Patrones léxicos.....	41
3.1.2	Patrones semánticos.....	42
3.1.3	Patrones sintácticos	45
3.2	Grafo de Interacción de eventos para eventos disparadores y de resultado	45
3.2.1	EIG. Grafo de interacción de eventos	46
3.2.2	Representación de eventos disparadores y de resultado	47
3.2.3	Representación de procesos.....	48
3.3	Reglas de consistencia	51
4. Prototipo.....		65
4.1	Interfaz gráfica	67
4.2	Funcionamiento	68
5. Validación del mecanismo		71
5.1	Diálogo controlado	71
5.2	Tarjetas de educación de requisitos	74
5.3	Esquema preconceptual	76
5.4	Diagrama de procesos.....	78
5.5	Tabla explicativa de procesos.....	79
5.6	Grafo de interacción de eventos, EIG	83
5.7	Diagrama de máquina de estados	84

Lista de Figuras

	Pág.
Figura 1. Notación del evento disparador en BPMN.	24
Figura 2. Notación del evento de resultado en BPMN.	25
Figura 3. Esquema general de los artefactos de UNC-Method	27
Figura 4. Fases y artefactos que representan eventos en UNC-Method	29
Figura 5. Meta-estructura del diálogo controlado	30
Figura 6. Tarjetas de educación requisitos.	31
Figura 7. Diagrama de procesos	34
Figura 8. Símbolos que se utilizan en un diagrama de máquina de estados.	36
Figura 9. Diagrama de máquina de estados	37
Figura 10. Notación del evento disparador y de resultado.	33
Figura 11. Esquema preconceptual (dominio ZenPhoto)	33
Figura 13. Diagrama de procesos de negocio.	39
Figura 14. Diagrama de máquina de estados	40
Figura 12. Partes del mecanismo.	41
Figura 16. ESG	46
Figura 15. EFG	46
Figura 17. EIG	47
Figura 18. Notación para eventos disparadores.	47
Figura 19. Notación para eventos de resultados	48
Figura 20. Grafo de interacción de eventos	50
Figura 21. Interfaz gráfica del prototipo.	67
Figura 22. Pregunta en el dialogo controlado sin evento.	68
Figura 23. Pregunta en el dialogo controlado con evento.	69
Figura 24. Evento disparador en el dialogo controlado.	69
Figura 25. Evento Condicional en el dialogo controlado.	70
Figura 26. Tarjetas de educación de requisitos.	75
Figura 27. Corrección Tarjetas de educación de requisitos.	75
Figura 28. Corrección de Eventos en diagrama preconceptual	76
Figura 29. Esquema preconceptual	77
Figura 30. Diagrama de procesos	78
Figura 31. Corrección de eventos en diagrama de procesos	79
Figura 32. Diagrama de procesos	83
Figura 33. Diagrama de máquina de estados	84
Figura 34. Corrección de eventos en diagrama de máquina de estados	84

Lista de Tablas

	Pág.
Tabla 1. Tipos de eventos disparadores básicos	24
Tabla 2. Tipos de eventos disparadores avanzados	24
Tabla 3. Tipos de eventos de resultado básico	26
Tabla 4. Tipos de eventos de resultado avanzado	26
Tabla 5. Tabla explicativa de procesos	36
Tabla 6. Síntesis de la revisión de literatura.....	16
Tabla 7. Clasificación de los verbos.....	42
Tabla 8. Definición de evento disparador tipo mensaje y señal.....	43
Tabla 9. Definición de eventos disparador tipo temporizador	44
Tabla 10. Representación general de los eventos disparadores	44
Tabla 11. Representación general del evento de resultado.....	45
Tabla 12. Reglas de consistencia para los eventos en artefactos de UNC-Method.....	52
Tabla 13. Tabla explicativa de procesos	80

Introducción

Los eventos son sucesos que ocurren en un tiempo preciso y proporcionan información relevante sobre el funcionamiento de procesos del sistema y la definición de los requisitos funcionales en un proyecto de software. Además, se encargan de generar cambios de estado en los procesos de negocio como la finalización y el inicio de actividades (Weinbach & García, 2004). Existen dos tipos de eventos: evento disparador y evento de resultado. El evento disparador es el suceso que desencadena un proceso y el evento de resultado es la ocurrencia que lo finaliza cuando se genera un resultado (Oracle, 2005). De esto modo, los eventos son importantes en el ciclo de vida del software porque influyen permanentemente en el flujo de los procesos de un proyecto de software.

UNC-Method es un método para la educación de requisitos de software (Zapata y Arango, 2006) que contempla cuatro etapas: contexto del software, análisis del problema, propuestas de solución y esquema conceptual. Con el desarrollo de estas etapas se describe el dominio del problema de una futura aplicación de software y se genera una solución a este problema. Así, los estudiantes de ingeniería de sistemas de la Universidad Nacional de Colombia utilizan UNC-Method para aprender los principios fundamentales de la educación de requisitos y practicar la consistencia del desarrollo del software (Zapata, 2012).

La consistencia es una de las principales características de calidad en el desarrollo de software (Zowghi & Gervasi, 2002), ya que permite la transición de una fase a otra en

el ciclo de vida del software y la articulación del problema con la solución (Zapata y Arango, 2006). Así, la consistencia implica la relación adecuada entre los artefactos y elementos que representan los problemas del dominio y su solución. En UNC-Method se encuentran inconsistencias en algunos artefactos que representan eventos. El dialogo controlado, las tarjetas de educación de requisitos, los esquemas preconceptuales, el diagrama de procesos y el diagrama de máquina de estados carecen de una representación completa de los eventos y su interacción. Por ello, se requiere un artefacto que represente la interacción entre los eventos y estructuras gramaticales que formalicen los eventos y los diferencien de las acciones y operaciones de un usuario.

En esta Tesis de Maestría se define un mecanismo de consistencia para los eventos disparadores y de resultado en el ciclo de vida del desarrollo de software en algunos artefactos de UNC-Method. El mecanismo consta de estructuras gramaticales para los eventos, una adaptación del grafo de interacción de eventos, un prototipo como apoyo al mecanismo de consistencia y la validación de la consistencia mediante algunos casos de estudio.

Conforme a lo anterior, esta Tesis comprende seis capítulos para la presentación del mecanismo de consistencia: en el Capítulo 1 se presenta el alcance y planteamiento del problema; en el Capítulo 2 se presenta el marco teórico para la caracterización del dominio; en el Capítulo 3 se propone el mecanismo de consistencia como solución al problema; en el Capítulo 4 se propone el mecanismo de consistencia como solución al problema; en el capítulo 5 se expone el prototipo como herramienta de apoyo al mecanismo propuesto; en el Capítulo 6 se valida la consistencia del mecanismo en algunos casos de estudio; finalmente, se presentan las conclusiones y el trabajo futuro.

1. Alcance y planteamiento del problema de investigación

1.1 Antecedentes

Los autores que se mencionan a continuación presentan trabajos en eventos asociados a las acciones que un usuario desempeña en la interfaz gráfica de usuario (ejemplo: dar clic a una opción de un menú: Singh *et al.* (2009) presentan una extracción automática de eventos desde especificaciones de requisitos textuales mediante un etiquetador. Los eventos que se utilizan hacen referencia a las acciones que ejecuta un usuario en la interfaz gráfica. Guerra y De Lara. (2007) presentan las gramáticas basadas en eventos, como gramáticas de grafos especialmente para entornos de modelado visual que generan metamodelado. Las reglas de estas gramáticas se puede disparar por las acciones del usuario como: La creación, edición o elementos de conexión.

Los siguientes autores tienen trabajos en consistencia: Vargas (2009) propone un método para establecer la consistencia de los problemas en el diagrama causa efecto con el diagrama de objetivos de Kaos. Pero en ningún momento hace referencia dominio de eventos. Zapata y Arango (2006) proponen algunas reglas de Consistencia para el Esquema Conceptual. Pero en ningún momento para los eventos disparadores y de resultado. Zapata (2012) expone la necesidad que hay respecto a la representación de eventos en los esquemas preconceptuales y la inconsistencia que se genera con el diagrama de procesos. La representación de los eventos en esquemas preconceptuales se basa en la que utiliza BPMN en diagramas de

procesos. A pesar de que existe un esfuerzo por tratar de representar los eventos, no se presenta consistencia en los artefactos que contienen eventos.

Los siguientes autores hacen referencia a los eventos que considera esta tesis, eventos disparadores y de resultado: Owen y Raj (2003) presentan una forma de especificación para los eventos con una representación que incluye mensajes, temporizadores y condiciones de error. Sin embargo no mencionan elementos de consistencia para los demás artefactos. Hernández y Álvarez (2010) indican que los eventos afectan los cambios del estado de los procesos y los flujos de la secuencia de actividades. Ellos presentan las definiciones de los elementos de notación de un diagrama de máquina de estados en un modelo de pruebas. Sin embargo no se encuentra un modelo de consistencia para los eventos en la representación total del sistema. Grgec y Muzâr (2007) revelan la importancia de los eventos en relación con los objetos dinámicos del sistema, llevándolos al diagrama de máquina de estados mediante la transición de un estado actual a otro. Samuel *et al.* (2008), Swinnen *et al.* (2010) y Lerner *et al.*, (2010) proponen una generación automática de casos de prueba utilizando diagramas de estado UML, en este trabajo se describen varios tipos de eventos pero no se representan en el artefacto.

En la Tabla 6 se presenta una síntesis de la revisión de la literatura especializada en eventos. Se pone de manifiesto la escasa participación que tiene la consistencia de eventos, puesto que, en general, se trata de trabajos sobre artefactos aislados sin establecer una manera de relacionar los eventos con otros artefactos. También, se presentan algunas falencias en la representación de los eventos de cualquier tipo y en la utilización de una gramática para la definición de eventos. Varias de las confusiones que se presentan en la literatura sobre el uso de operaciones y funciones como si fueran eventos se deben a la carencia de una gramática formal que apoye la definición de los eventos. Las deficiencias que se compendian en la Tabla 6 posibilitan la

definición del alcance y el planteamiento del problema de investigación que da origen a esta Tesis de Maestría y que son objeto de este Capítulo.

Tabla 1. Síntesis de la revisión de literatura

AUTORES	Enfoques						
	Evento	Evento disparador	Evento de resultado	Artefactos	Gramática	Consistencia de artefactos	Consistencia de eventos
Arango y Zapata (2006)	X	X	X			X	
Ben, Y., Ben L. (2009)	X				X		
Eloranta, L., Kallio E and Terho I.(2006)	X						
Fai, K. and Nicolici, N. (2008)	X	X					
Grgec, M. and Muzár, R. (2007)	X						
Guerra, E. and Lara J. (2007)	X			X	X		
Hernández, U. and Álvarez F. (2010)	X	X	X				
Leonardi, M. and Felice, L. (2009)	X	X					
Lerner, B. and Christov, S. (2010)	X	X	X				
Owen, M. and Raj, J. (1993)	X	X	X	X			
OMG. (2009)	X	X	X	X			
Samuel, P., Mall, R. and Bothra. (2003)	X	X	X	X	X		
Schwitters, R. (2008)	X				X		
Singh, S.K., Gupta, R., Sabharwal, S. and Gupta, J.P (2011)	X		X				
S. Sabharwal. S. Kumar, D. and Sabharwal. (2009)	X	X	X				
Stephen, K., Wood, H., Oleg, W., Gareth, J., Howells and Klaus, D. (2008)	X	X		X			
Vargas, F (2010)						X	
Zapata, C. (2012)	X	X	X	X	X	X	
Zapata C. y Lezcano, L. (2007)	X				X		
Zapata, C. Villegas, S. and Arango, F. (2006)	X	X	X	X			

1.2 Problema

El método de educación de requisitos UNC-Method comprende etapas que permiten describir detalladamente el dominio del problema de una futura aplicación de software, mediante la utilización de diferentes artefactos (Zapata y Arango, 2006). Algunos de estos artefactos (el dialogo controlado, las tarjetas de educación de requisitos, el esquema preconceptual, el diagrama de procesos y el diagrama de máquina de estados) incluyen la representación de eventos disparadores y de resultado. La representación de eventos en UNC-Method es importante y necesaria, ya que estos permiten definir requisitos funcionales y dan información relevante acerca del funcionamiento del sistema (Weinbach y García, 2004). Además, afectan los cambios del estado de los procesos y los flujos de la secuencia de las actividades (Hernández & Alvaréz, 2010). A pesar de que en UNC-Method se evidencia la representación de los eventos disparadores y de resultado, aún falta plenitud en esta representación, ya que no existen reglas que permitan mantener la consistencia de estos eventos entre los artefactos que los representan durante todas las etapas del método. Adicionalmente, no existen artefactos que representen la interacción entre los eventos, para observar características como: relación, secuencia y orden de ocurrencia de los eventos. Finalmente, los eventos se suelen confundir con acciones y operaciones que ejecutan los usuarios (Osorio, 2008), provocando así, una representación errónea de los eventos.

De acuerdo con lo anterior, se presenta la necesidad de proponer un mecanismo de consistencia para los eventos disparador y de resultado en los artefactos de UNC-Method permita la solución de los problemas expuestos.

1.3 Objetivos

1.3.1 General

Proponer un mecanismo de consistencia para los eventos disparadores y de resultado en los artefactos del método de educación de requisitos UNC-Method.

1.3.2 Específicos

- Identificar los artefactos de UNC-Method que requieren la representación de los eventos disparadores y de resultado.
- Identificar las características que permitan una representación completa de los eventos disparadores y de resultado.
- Establecer la estructura gramatical de los eventos representados en los artefactos de UNC-Method.
- Definir un mecanismo que permita representar los eventos disparador y de resultado en los artefactos de UNC-Method empleando reglas de consistencia, que garanticen la articulación de los diferentes artefactos con el discurso del interesado.
- Elaborar un prototipo para la aplicabilidad del mecanismo en los artefactos de UNC-Method.
- Validar el mecanismo propuesto en un caso de estudio que ejemplifique la representación de los eventos disparadores y de resultado en los artefactos de UNC-Method.

1.4 Hipótesis

Al plantear un mecanismo de consistencia para los eventos disparador y de resultado en los artefactos de UNC-Method, se garantiza una completa representación de los eventos, la relación, secuencia y orden de ocurrencia que debe existir entre los

eventos de los diferentes diagramas que lo representan, permitiendo un mejor análisis del dominio problema y efectivas propuestas de solución que respondan a las necesidades del interesado.

1.5 Motivación

La consistencia se concibe en UNC-Method como un aspecto fundamental que permite la articulación de una fase con otra en el desarrollo de un proyecto de software y precisa la coherencia interna entre los artefactos de la solución (Zapata y Arango, 2006), es decir la completa representación de los elementos entre los artefactos. La consistencia se convierte en insumo para la calidad en la ingeniería de software como lo describen Zowghi y Gervasi (2002), de esta manera no solo le concierne al método o al proceso de educación de requisitos sino a todo el proceso del ciclo de vida del software.

La definición de un mecanismo de consistencia para los eventos disparadores y de resultados en UNC-Method contribuye al mejoramiento de la consistencia en el tópico de eventos permitiendo determinar el comportamiento de los procesos en un proyecto de software, mejorar la comprensión del problema y el contexto y dar mejores soluciones al interesado.

La determinación de estructuras gramaticales, reglas de consistencia y un grafo de interacción de eventos en el mecanismo permiten la representación completa de los eventos disparadores y de resultado en algunos artefactos de UNC-Method.

1.6 Metodología

1.6.1 Problema de investigación

Esta fase comprende 3 etapas: La primera, comprende la caracterización de eventos, con el objetivo de observar su representación en artefactos de software.

En la segunda etapa se realiza una exploración del método de la Universidad Nacional, UNC-Method. Se efectúa la Identificación de los artefactos de UNC-Method que requieren la representación de los eventos disparadores y de resultado.

En la tercera etapa, se estudian otros artefactos que especifiquen como solucionan el problema de consistencia en sus artefactos.

En esta fase se atienden los objetivo específico 1 y 2.

1.6.2 Análisis

Con el desarrollo la fase de exploración de la revisión de literatura de los eventos disparadores y de resultados, se procede al análisis específico de las características que permiten una representación completa de los eventos disparadores y de resultado.

El objetivo de esta fase es que a partir de la literatura revisada, realizar una análisis exhaustivo para lograr identificar la representación de cada uno de los artefactos de UNC-Method y realizar una relación que establezca coherencia entre ellos.

En esta fase se atienden los objetivos específicos 1 y 2.

1.6.3 Diseño de la solución

El objetivo de esta fase, es que a partir del análisis de las características de los eventos disparador y de resultado se puedan representar en los artefactos de UNC-Method mediante la creación de un mecanismo de consistencia que permita una representación completa en artefactos del método que contengan eventos.

Esta fase se divide en tres etapas: La primera consiste en definir estructuras gramaticales de los eventos disparadores y de resultado. En la segunda fase se pretenden establecer reglas de consistencia para los eventos mencionados y en la tercera se pretende elaborar un artefacto que relacione la interacción entre los eventos. Además se pretende definir un prototipo que permita obtener la aplicabilidad de aspectos del mecanismo en artefactos de UNC-Method.

Esta fase atiende a los objetivos específicos 3, 4 y 5.

1.6.4 Validación de la solución

En esa fase, se elige un caso de estudio que ejemplifique la representación de los eventos disparadores y de resultado en los artefactos de UNC-Method de la validación del método de reglas de consistencia propuesto.

Esta fase atiende a los objetivos específicos 3, 4, 5 y 6.

2. Marco teórico

2.1 Marco referencial

2.1.1 Eventos

El concepto de *evento* se utiliza en el ciclo de vida del desarrollo de software para indicar un suceso en un tiempo y lugar específico que puede iniciar o finalizar un proceso del sistema. La caracterización y el análisis de eventos son importantes para el proceso de educación de requisitos, ya que los eventos desencadenan procesos provocando cambios en su estado y sus relaciones (Singh *et al.*, 2009). Los eventos son fundamentales en la creación de requisitos funcionales debido a que representan acontecimientos importantes en los procesos de negocio, como: el inicio o finalización de una operación. De este modo, proporcionan información relevante sobre el comportamiento de los procesos en un producto de software (Weinbach y García, 2004). Los eventos, también, especifican los hechos externos o condiciones internas del sistema que determinan el inicio de un conjunto de procesos que se asocian empleando relaciones de precedencia (Oracle, 2005).

En el modelado de procesos de negocio BPMN (*Business Process Modeling Notation*) se utiliza un evento para referenciar algo que sucede en la ejecución de los procesos de negocio y que afecta los flujos de estos procesos, generando una causa o un impacto (Owen & Raj, 2003). En un proceso de negocio, un evento representa varias ocurrencias: el inicio de una actividad, el final de una actividad, el cambio de estado de un proceso y un mensaje que llega, entre otros. Sin embargo, la notación BPMN limita

el uso de eventos para incluir sólo los tipos de eventos que afectan la secuencia de tiempo de las actividades de un proceso.

En la programación orientada a eventos, generalmente asociada con la interfaz gráfica de usuario (GUI, por sus siglas en inglés), se relaciona el término *evento* con funciones y operaciones que el usuario realiza, como hacer un *click*, elegir una opción o pulsar una tecla (Osorio, 2008). En consecuencia, en el contexto del ciclo de vida del desarrollo de software el término *evento* se suele confundir con operaciones y funciones como se utiliza en las GUI. Sin embargo, cuando se trata de operaciones o funciones existe un sujeto que ejecuta la acción, mientras que el concepto de *evento* que se utiliza en los procesos de negocio expresa un suceso que ocurre sin que un sujeto lo realice.

2.1.2 Tipos de eventos

▪ Eventos disparadores

Se denomina *evento disparador*, al evento que inicia uno o más procesos. (Oracle, 2005). Estos eventos desencadenan los procesos como consecuencia de una causa o un disparo (Hernández & Alvarez, 2010). Los eventos disparadores indican el punto de partida de los procesos y establecen el orden de su ocurrencia. Asimismo, sólo tienen flujos de secuencia de salida y pueden, también, desencadenar la ejecución de otros eventos disparadores (OMG, 2009).

Los eventos disparadores coinciden con el surgimiento de alguna necesidad, la llegada de una fecha o situación particular, por ejemplo el fin de un mes o la llegada de un pedido (Zapata y Arango, 2006). De este modo, un evento disparador puede ser una declaración o una restricción. Como declaración, el evento disparador es la instrucción que hace que se genere un disparo; como restricción especifica una condición que se evalúa como verdadera para disparar un proceso (Oracle, 2005).

En la notación BPMN se define como evento disparador el que inicia el flujo o la secuencia de un proceso. Este evento se representa gráficamente con un círculo en línea delgada (véase la Figura 1), dentro del cual se pueden colocar marcadores para mostrar variaciones del evento.

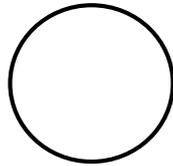


Figura 1.Notación del evento disparador en BPMN.

En BPMN se representan seis tipos de eventos disparadores, cuatro básicos (simple, mensaje, temporizador y señal) y dos avanzados (condicional y múltiple), que se describen en las Tablas 1 y 2 (OMG, 2009).

Tabla 2. Tipos de eventos disparadores básicos

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.	
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.	

Tabla 3. Tipos de eventos disparadores avanzados

Evento	Descripción	Símbolo
Condicional	Es un disparador con una expresión de condición que se debe satisfacer para que empiece el proceso.	
Múltiple	Hay uno o más disparadores (mensajes, condiciones, señales o temporizadores) que pueden iniciar un proceso.	

▪ Eventos de resultado

Se define el *evento de resultado* como la consecuencia que se da después de la culminación de uno o más procesos (Oracle, 2005). Los eventos de resultado frecuentemente se expresan en términos de verbos en participio pasado y se pueden relacionar con eventos disparadores. Por ejemplo, el evento de resultado *orden diligenciada* se puede relacionar con el evento disparador *llega una orden* (Zapata y Arango, 2006).

En la notación BPMN se utilizan los eventos de resultado en el flujo de secuencia de los procesos, como los que finalizan el flujo de un proceso. Por ejemplo se envía un mensaje o se transmite una señal (White & Derek, 2009). Se representan gráficamente con círculos en línea gruesa. Al igual que en los eventos disparadores, puede variar su marcador (OMG, 2009).

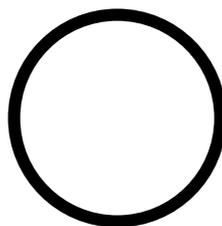


Figura2. Notación del evento de resultado en BPMN.

Los eventos de resultado se agrupan según su tipo en cuatro básicos y cuatro avanzados. Los básicos son simple, mensaje, señal y terminador y los avanzados son error, cancelación, compensación y múltiple. En la Tablas 3 y 4 se puede apreciar la caracterización de cada uno. (OMG, 2009).

Tabla 4. Tipos de eventos de resultado básico

Evento	Descripción	Símbolo
Simple	No se define ningún resultado.	
Mensaje	Comunicación con otra entidad de negocio. (Participante o proceso).	
Señal	Define un evento que envía información a uno o más receptores el cual cualquier otro proceso puede ver y al cual puede reaccionar.	
Terminador	Suspende todas las actividades de un proceso.	

Tabla 5. Tipos de eventos de resultado avanzado

Evento	Descripción	Símbolo
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 subproceso 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.	

2.1.3 UNC-Method

Arango y Zapata (2006) crearon UNC-Method como un método para la educación de requisitos de software. Este método se enfoca en la descripción cuidadosa del dominio del problema de una futura aplicación de software para generar una aplicación como solución al problema identificado en el dominio. El UNC-Method contempla cuatro etapas que se esquematizan en la Figura 3: contexto del software, análisis del

problema, propuestas de solución y esquema conceptual. El contexto del software caracteriza la organización, con sus funciones y relaciones de poder, los actores y sus responsabilidades y el vocabulario del área. El análisis del problema representa los procesos de la organización, los objetivos asociados con esos procesos y los principales problemas y sus causas. Las propuestas de solución permiten identificar un conjunto de soluciones entre las cuales se elige la mejor. Por último, en el esquema conceptual se representa la solución seleccionada empleando alguna lógica de primer orden, se determina la estructura de la solución y se modela el comportamiento que pueden exhibir algunos objetos y algunas transacciones de la solución. Los estudiantes de ingeniería de sistemas de la Universidad Nacional de Colombia utilizan el UNC-Method para aprender los principios fundamentales de la educación de requisitos y practicar la consistencia del desarrollo del software (Zapata, 2012).

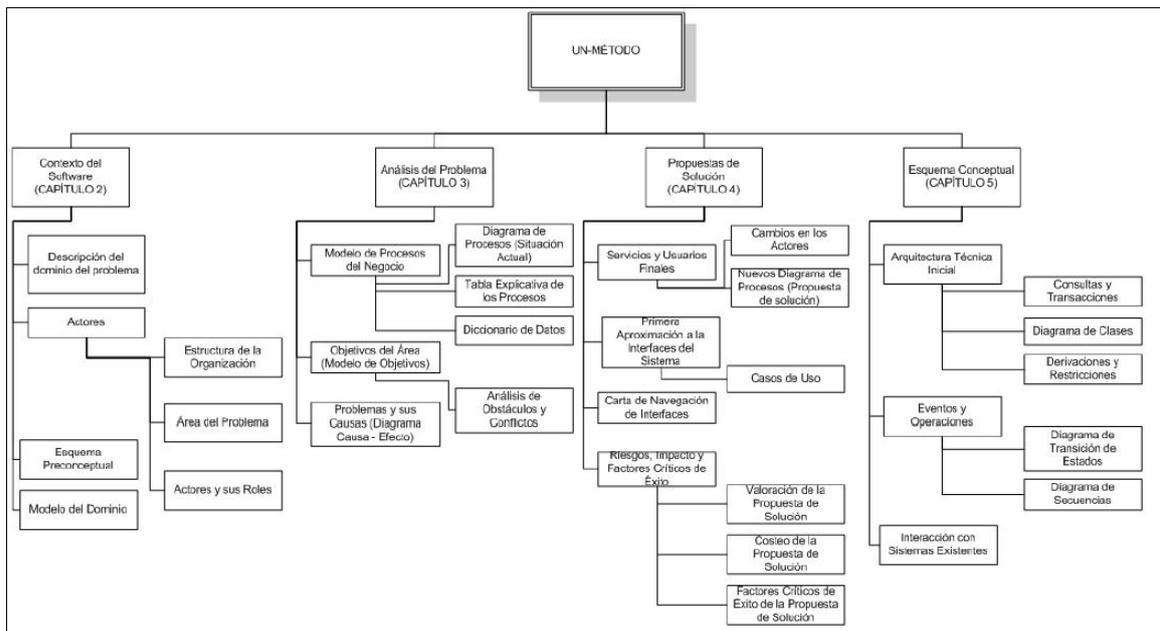


Figura 3. Esquema general de los artefactos de UNC-Method

2.1.4 Consistencia

En ingeniería de software, se entiende por consistencia la relación que debe existir entre los elementos de los diferentes artefactos que componen una descripción del dominio (Zowghi & Gervasi, 2002). Esta relación, es una característica fundamental que permite la articulación de una fase con otra en el desarrollo de un proyecto de software y precisa la coherencia interna entre los artefactos de la solución. De acuerdo con lo anterior, mediante la consistencia se entregan soluciones que satisfacen las expectativas y necesidades de los interesados.

En UNC-Method los aspectos de consistencia son esenciales, ya que hacen referencia a la misma utilización de los conceptos y relaciones en los artefactos y elementos que requiere el método. La consistencia se establece mediante reglas que sirven para verificar la coherencia que deben mantener los diferentes artefactos.

2.1.5 Mecanismo

Un mecanismo es, en términos generales, un conjunto de elementos que cumplen una función para lograr un fin específico, sin determinar el orden de ocurrencia. Este conjunto de elementos posibilita el ajuste de los demás, permitiendo que todos cumplan su función. En consecuencia, la noción de mecanismo permitirá el cumplimiento de un objetivo sistémico. (Gerring, 2007).

2.2 Dominio del problema

2.2.1 Representación de eventos en artefactos de UNC-Method

Los artefactos que representan eventos se analizan en la verificación de los problemas de consistencia de UNC-Method. En consecuencia, a continuación se especifican las fases y artefactos que se tuvieron en cuenta en el problema y solución de esta Tesis de Maestría (véase la Figura 4).

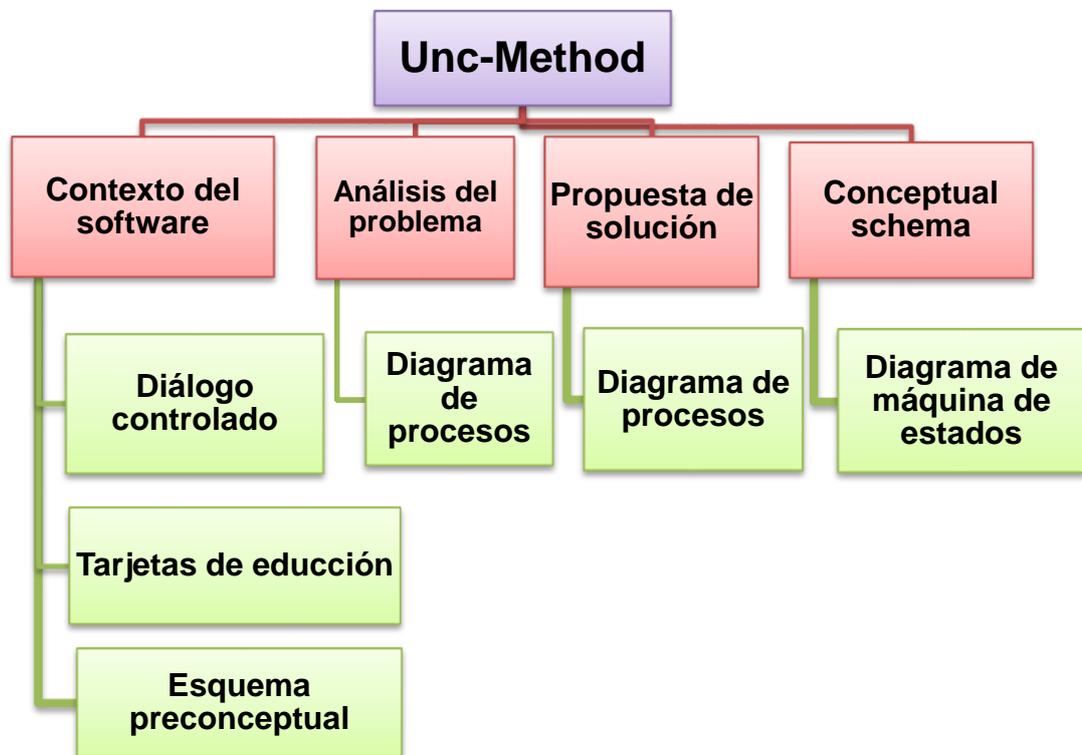


Figura 4. Fases y artefactos que representan eventos en UNC-Method

2.2.2 Diálogo controlado

Zapata y Giraldo (2009) exploran la posibilidad de restringir las respuestas de los interesados (*stakeholders*) a las preguntas que realizan los analistas en el diálogo de educación de requisitos, controlando las listas y oraciones para la especificación de los entregables. Además, intentan usar estas respuestas de los interesados como

entradas para obtener automáticamente un esquema preconceptual que represente el contexto del software.

Zapata y Carmona (2010) presentan una formalización del modelo de diálogo para la educación de requisitos y una ejemplificación del tipo de preguntas y respuestas que se deben realizar. En la Figura 5 se sintetizan los elementos que debería contener el dialogo controlado.

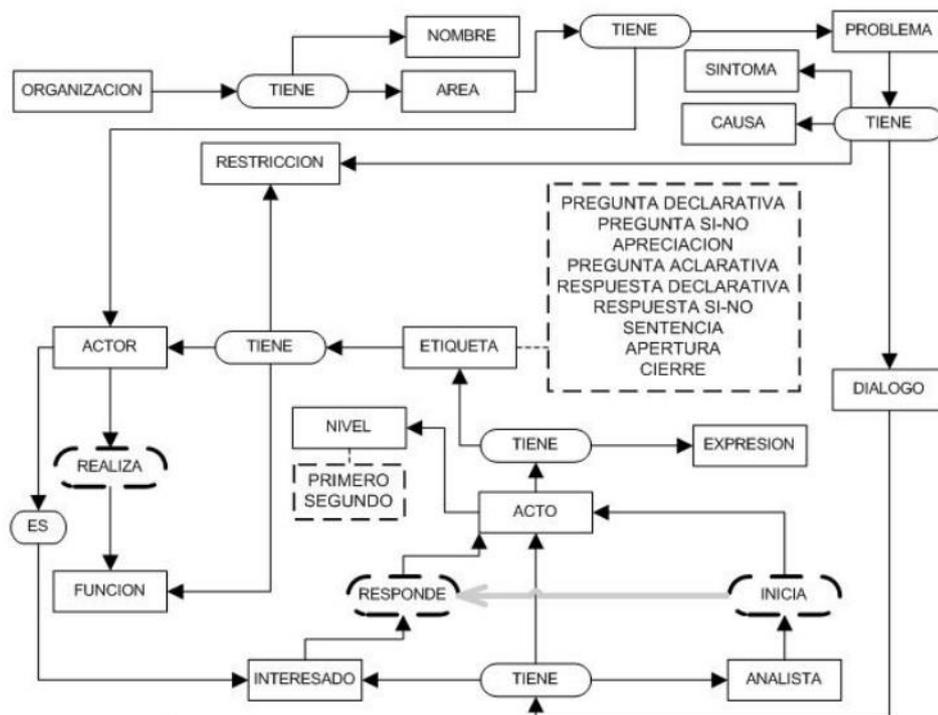


Figura 5. Meta-estructura del diálogo controlado

Algunos ejemplos de las preguntas y respuestas que se relacionan con eventos se plantean a continuación:

Analista: ¿Qué requiere el usuario para lograr la subida de un archivo?

Interesado: El sólo necesita que aparezca un archivo.

En este sentido las preguntas y respuestas siempre inician de la misma manera, cambia la frase que sigue, de acuerdo con el dominio. Se puede identificar que la respuesta que da el interesado (*aparezca un archivo*) representa un evento disparador ya que menciona un verbo que se relaciona con estos (*aparecer*).

En el caso de un evento condicional utilizando la misma estructura se puede observar que se presentan cambios en la respuesta del interesado, como ocurre en el siguiente diálogo:

Analista: ¿Qué requiere el usuario para lograr editar una etiqueta?

Interesado: El sólo necesita que tenga descripción de derechos de autor = “Admin” y descripción de derechos de autor = “Etiquetar”. De esta manera no se presentan reglas precisas para los eventos en el diálogo controlado.

2.2.3 Tarjetas de educación de requisitos

Arredondo (2009) trabajó en tarjetas, como artefacto para asistir el proceso de educación de requisitos, donde se muestra las características de los actores y sus funciones. En las tarjetas de actores se plantean sus características y en las tarjetas de funciones se evidencian el objeto y restricciones acompañadas de los objetivos y problemas asociados. Estas restricciones contienen respuestas de los interesados que se asocian con eventos que, al parecer, son eventos disparadores pero no se describen en ninguna parte (véase la Figura 6).

ACTOR		ANOTACIONES	
Usuario			
Características	Datos personales, Clave, Nombre, email, derechos, nombre_usuario, grupo, acceso, id		
FUNCIÓN		OBJETIVO	PROBLEMA
Subir		1. Fomentar archivos los 2. Fomentar permisos los	Es difícil lograr subir los archivos
ACTOR	Usuario		
OBJETO	Archivo		
RESTRICCIÓN			
1. Descripción de los derechos de autor= “Admin” y descripción de los derechos de autor= “Archivo” 2. Aparece archivo			

Figura 6. Tarjetas de educación requisitos

2.2.4 Esquemas preconceptuales

Zapata y Arango (2006) definen los esquemas preconceptuales como una representación intermedia entre las especificaciones textuales en lenguaje natural y los diferentes esquemas conceptuales que permiten el modelado de una aplicación de software. Los esquemas preconceptuales permiten representar de manera controlada conceptos, relaciones y restricciones del discurso del dominio del interesado. Zapata (2012) introduce los eventos y su notación en los esquemas preconceptuales, con base en la notación que utiliza BPMN (*Business Process Modeling Notation*) en los diagramas de procesos de negocios: un círculo como símbolo del evento. Se utiliza el evento disparador para iniciar una relación dinámica y el evento de resultado para finalizar las relaciones dinámicas (en este caso, se describe con un verbo en participio pasado). En los eventos disparadores, se utiliza una relación eventual que usa un concepto y un verbo intransitivo y se representa con un ovalo en doble línea como se observa en la Figura 7. Zapata (2012) presenta la necesidad que hay respecto a una representación completa de eventos en los esquemas preconceptuales y las inconsistencias que se generan al pasarlos al diagrama de procesos. También, indica que se requiere adicionar información relacionada con los eventos en algunos artefactos de UNC-Method. Adicionalmente, los eventos disparadores se pueden representar como condicionales como se puede apreciar en la Figura 8 y no existen reglas de consistencia que permitan definirlos en los diferentes artefactos.

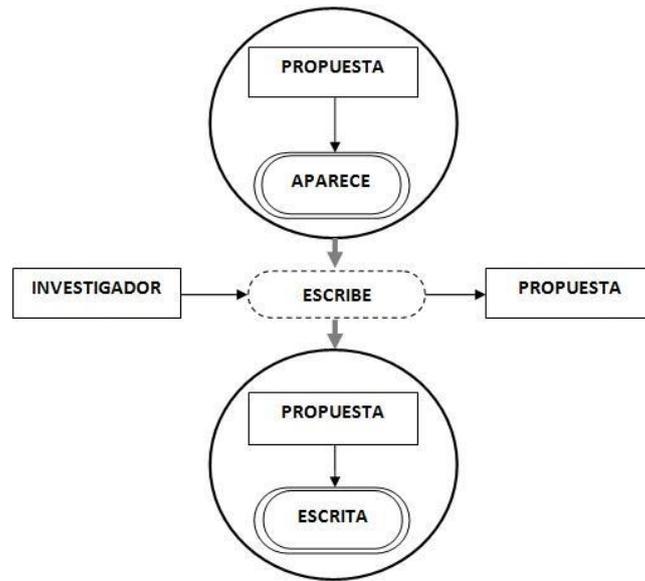


Figura 7. Notación del evento disparador y de resultado.

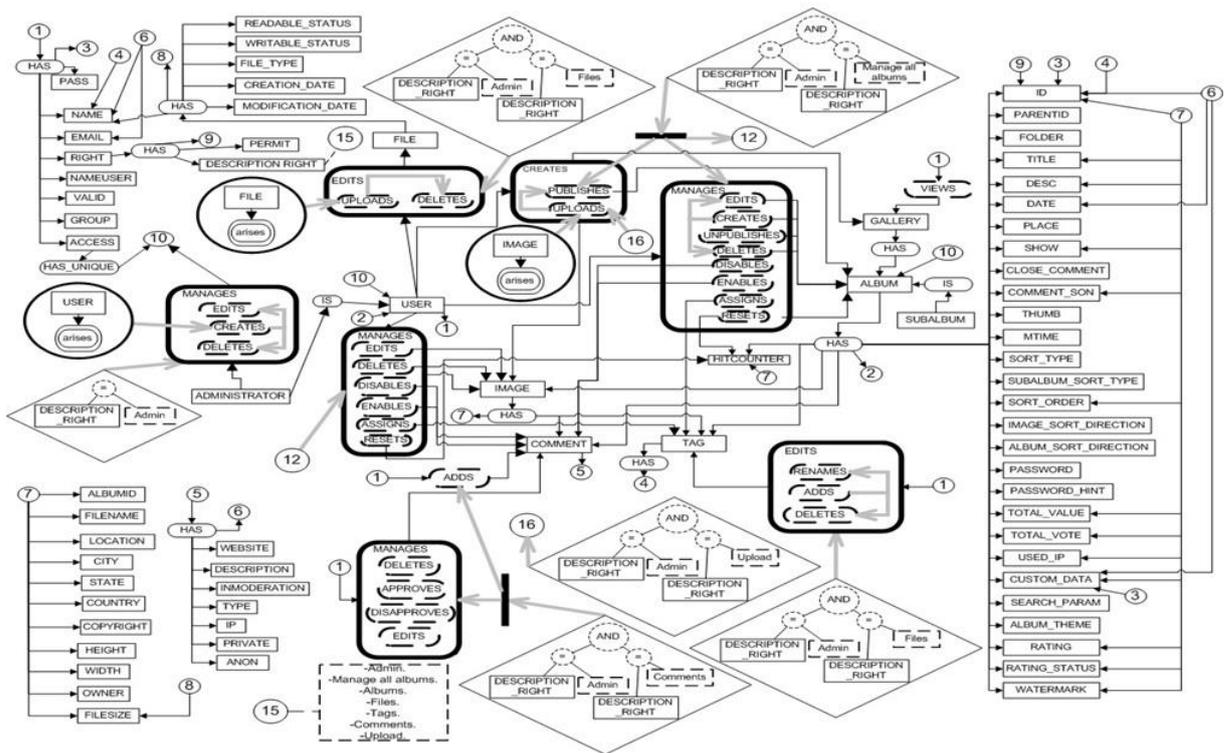


Figura 8. Esquema preconceptual (dominio ZenPhoto)

2.2.5 Diagrama de procesos

El diagrama de procesos que se utiliza en el UNC-Method se fundamenta en la representación del diagrama de procesos de CDM (*Custom Development Method*) que relaciona los procesos, almacenes, eventos y condicionales. En el diagrama de procesos de UNC-Method se utiliza la flecha de entrada como representación para los eventos disparadores hacia un proceso y se les coloca un código que comienza con TE. Adicionalmente, se usa la flecha de salida después de un proceso como representación para los eventos de resultado, con su respectivo código que inicia con RE, como se observa en la Figura 9 (En este artefacto “S” representa un almacenamiento y “P” un proceso). De acuerdo con esta mención, en el diagrama de procesos se observa una clara notación gráfica para los eventos. Sin embargo, la información adicional que se ingresa en las flechas de evento no presenta unas reglas de formación precisas que permitan darle consistencia; se suele usar el nombre del proceso en participio pasado para los eventos de resultado y no hay sugerencias para los eventos disparadores (Zapata, 2012). Para complementar la información del diagrama, se emplea un artefacto denominado *tabla explicativa del diagrama de procesos* (véase la Tabla 6).

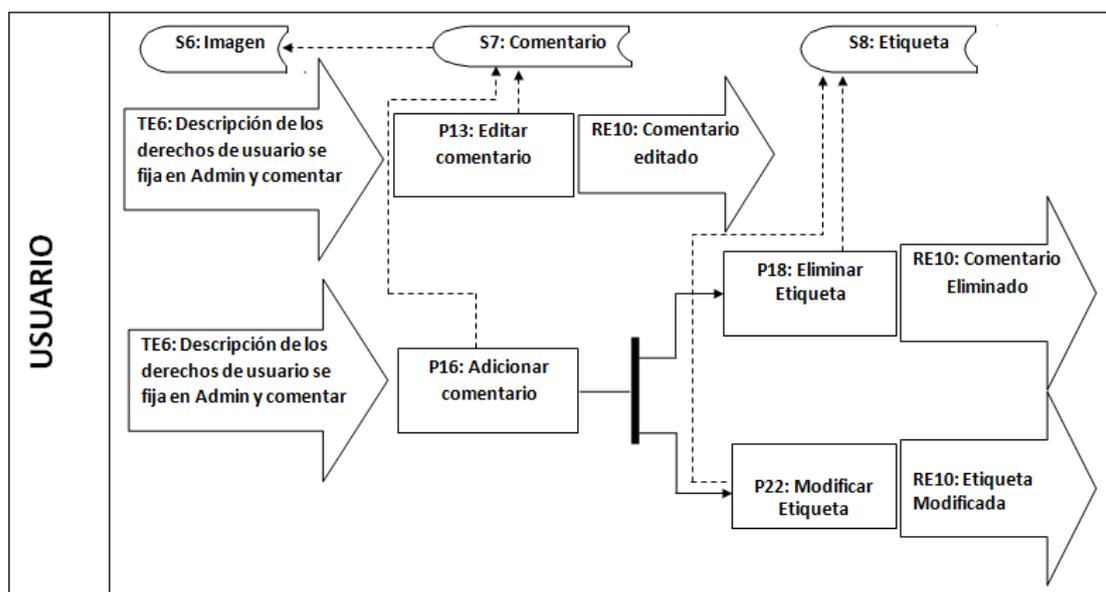


Figura 9. Diagrama de procesos

2.2.6 Tabla explicativa del diagrama de procesos

Esta tabla acompaña al diagrama de procesos para complementar información importante de los elementos que se relacionan con los procesos. A continuación se definen las columnas que contiene la tabla:

Nombre: El proceso se nombra con la unión de una relación dinámica con un concepto. Se sugiere colocarle un código que comienza con P.

Objetivo: Cada proceso se debe relacionar con al menos un objetivo cuyo código inicia con T, también, puede tener uno o varios requisitos asociados que se codifican comenzando con R.

Duración/frecuencia: Duración es el tiempo que toma completar un proceso. Frecuencia es el número de veces en que el tiempo se completa en cierto período.

¿Cómo?/ ¿Dónde?: *Cómo* es una corta descripción del proceso y *dónde* es el lugar donde se ejecuta el proceso.

Problemas: Se mencionan los problemas que afectan el logro de los objetivos, incluyendo un código que inicia con C.

Reglas de negocio: De manera opcional, si el proceso se relaciona con una regla de negocio se puede mencionar en esta columna.

Eventos: como se especificó previamente para los eventos disparadores se utiliza códigos que comienzan con TE. Los códigos de los eventos de resultado comienzan con RE.

En la *Tabla 5* se presenta un ejemplo de la tabla explicativa utilizando procesos en el dominio de una red social. En la tabla explicativa de procesos se utilizan los mismos códigos que en el diagrama de procesos, iniciando con TE para los eventos disparadores y RE para los eventos de resultado. No obstante, la reglas de consistencia se basan en pasar el mismo nombre en la tabla, pero no se especifica la manera correcta de definirlos.

Tabla 6. Tabla explicativa de procesos

Nombre	Objetivo	Duración/frecuencia	¿Cómo?/ ¿Dónde?	Problemas	Reglas de negocio	Eventos
P12: Eliminar imagen	G4: Fomentar los permisos R6: Hacer que la imagen pueda se pueda gestionar.	2 minutos; 20 veces a la semana	Un mal comentario y etiqueta son el origen de este proceso. Toma lugar en el usuario conectado.	C10: Hay pocos permisos para editar imágenes o álbumes	BR001: Administrar y manejar todos los álbumes BR006: Administrar	TE5: Descripción de derechos de autor se fija en Administrador y manejo de todos los álbumes. RE9: Imagen eliminada
P13: Editar comentario	G4: Fomentar los permisos. R10: Hacer que los comentarios se puedan gestionar.	4 minutos; 30 veces a la semana	Algunos errores en el comentario son el origen de este proceso. Toma lugar en el usuario conectado.	C3: Los permisos no son claramente definidos	BR004: Administrar y comentar; BR006: Administrar	TE6: Descripción de derechos de usuario se fija en Administrador y comentar RE10: Comentario editado.
P14: Asignar etiqueta	G4: Fomentar los permisos, R8: permite editar etiqueta.	3 minutos; 1500 veces a la semana	Este proceso da algunas explicaciones de usuarios. Toma lugar en el usuario conectado.	C3: Los permisos no son claramente definidos	BR001: Administrar y manejar todos los álbumes; BR006: Administrar	TE5: Descripción de derechos de autor se fija en Administrador y manejo de todos los álbumes. RE11: Etiqueta asignada.

2.2.7 Diagrama de Máquina de estados

Zapata (2012) muestra en UNC-Method el diagrama de estados, donde incluye los estados de los objetos importantes en el dominio. Utiliza los elementos y la sintaxis que definió el OMG (*Object Management Group*): estados, transiciones, estado inicial y estado final. Los estados son situaciones en la vida de un objeto que satisfacen una condición, esperan una condición o realizan una actividad. En esta estructura no se evidencian los estados que se emplean en UNC-Method. En la Figura 10 se muestra los símbolos que se utilizan en el diagrama y en la figura 11 un ejemplo de su uso.



Figura 10. Símbolos que se utilizan en un diagrama de máquina de estados.

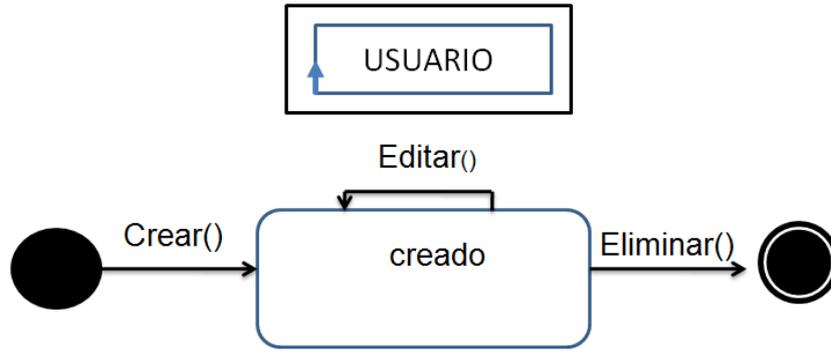


Figura 11. Diagrama de máquina de estados

3.Solución propuesta: Mecanismo de consistencia

El mecanismo de consistencia permite la representación completa de eventos disparadores y de resultado en artefactos de UNC-Method, mediante el cumplimiento de las funciones de sus tres partes: las estructuras gramaticales, las reglas de consistencia para los eventos y el grafo de interacción de eventos.

Para el planteamiento de la solución se tuvieron en cuenta dos artefactos que contienen de forma detallada eventos disparadores y de resultado que no se emplean en UNC-Method, El diagrama de procesos de negocio BPMN y el diagrama de máquina de estados de Fowler (2003).

- **Diagrama de procesos de negocio BPMN**

El OMG (2009) especifica la representación de los eventos en BPMN (*Business Process Modeling Notation*) durante el modelado de procesos de negocio, que 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. Owen, Raj (2003) y Eloranta *et al.* (2006) identifican que, a menudo, ocurre un evento mientras que un proceso en particular se está realizando, causando una interrupción en el proceso o provocando 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 se denota con un icono representativo. Sin embargo, no incluye reglas de consistencia que permitan llevar esta representación hacia otros artefactos. Lerner *et al.* (2010) especifican la utilización y representación de eventos por medio del estudio de manejo de excepciones BPMN, en diagramas de actividades y procesos. Se ofrece una visión general de los eventos disparadores usados en la definición de excepciones que incluye: mensajes de cancelación, múltiples disparos, error y señal. Sin embargo, en la representación sólo se modela el flujo de excepciones procedentes de los eventos. Un ejemplo de la notación BPMN incluyendo el manejo de eventos se muestra en la Figura 12.

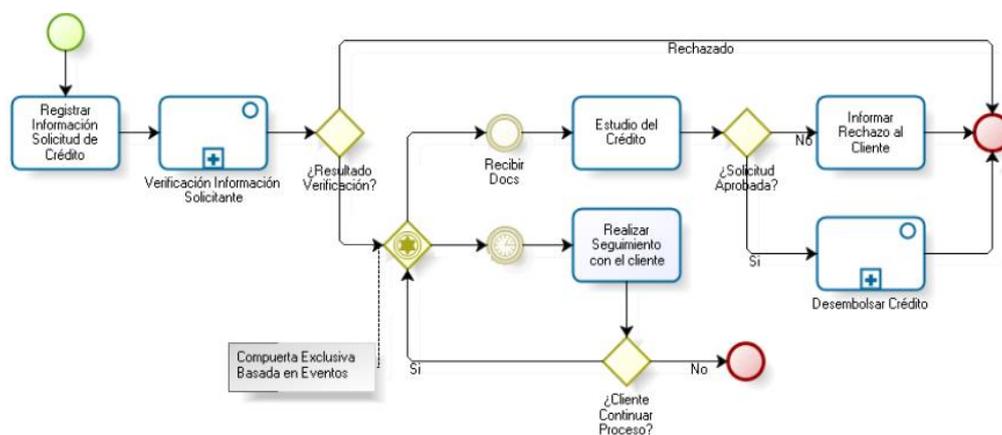


Figura 12. Diagrama de procesos de negocio

▪ Diagrama de máquinas de estado UML

Fowler (2003) define las transiciones en un diagrama de máquina de estados, como aquellas que trazan el camino entre los estados mediante intervalos de tiempo. Estas transiciones poseen la siguiente estructura: *Evento disparador [condición de guarda] acción*. El evento disparador provoca un cambio de estado en el objeto, la condición de guarda evalúa el momento de recibir el evento y habilita o no la transición de estado y la acción es el comportamiento que se ejecuta en la transición. El estado inicial indica el momento donde empieza un estado y el estado final el momento en que termina. A pesar de que los autores se esmeran en dividir tres elementos en la estructura de la transición de estados, se puede observar que la definición de los eventos de un evento disparador se establece como la de un evento de resultado, ya

que lo definen como un verbo en participio pasado (véase la Figura 13) *candado removido*, cuando debería ser *remueve candado*, que es realmente una operación.

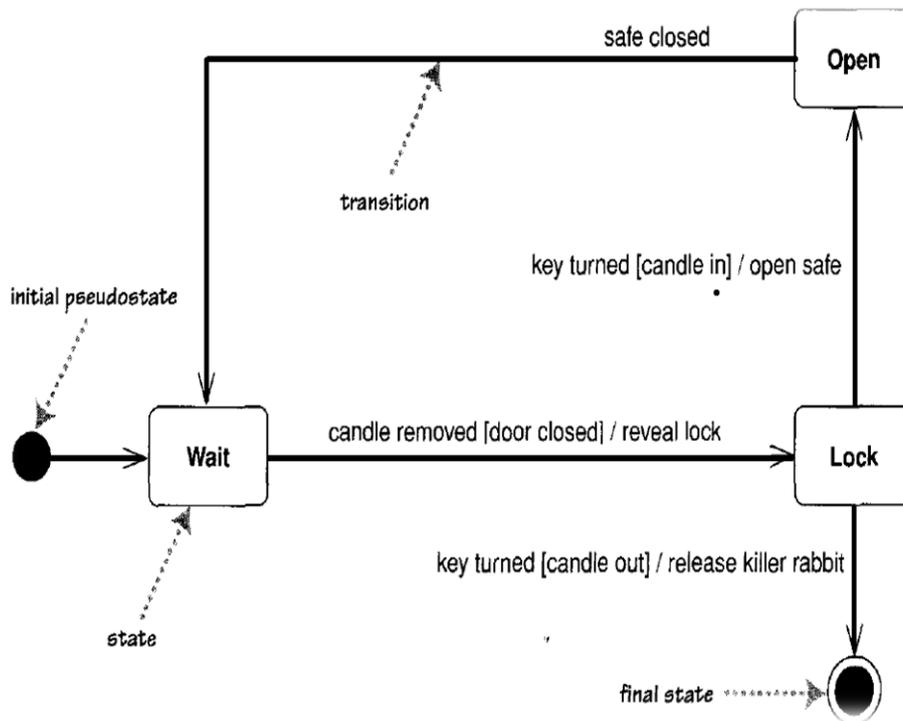


Figura 13. Diagrama de máquina de estados

En la Figura 14 se puede inferir que todas las funciones de las partes del mecanismo (estructuras gramaticales, reglas de consistencia y grafo de interacción de eventos) en conjunto permiten lograr consistencia entre los eventos que se representan en los artefactos de UNC-Method. En los siguientes puntos se describe el funcionamiento de cada una de cada una de sus partes.

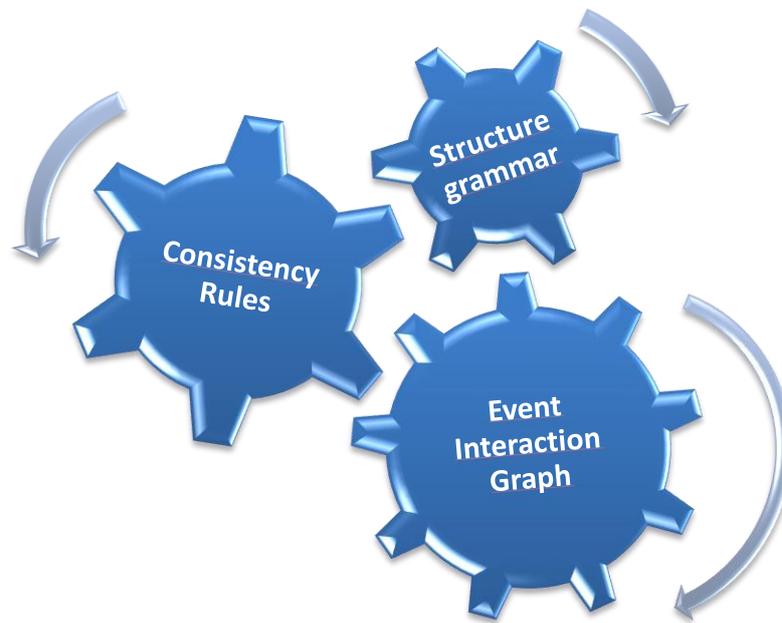


Figura 14. Partes del mecanismo

3.1 Estructuras gramaticales de los eventos

La estructura gramatical se refiere a la definición de los componentes lingüísticos que deben tener los eventos. Para el planteamiento de esta estructura se definen, a continuación, patrones lingüísticos para los eventos disparadores y de resultado que permiten una adecuada representación en los artefactos de UNC-Method y se esclarecen en detalle en la definición de las reglas de consistencia para los eventos.

3.1.1 Patrones léxicos

Hoey (1991) afirma que los patrones léxicos resultan relevantes, por cuanto permiten establecer restricciones en las funcionales gramaticales. Para la definición de patrones léxicos, los verbos constituyen un instrumento conceptual útil para identificar los factores semánticos que tienen efectos sobre la forma sintáctica con la que se expresa un evento. Se define un verbo como la representación de una situación o un estado con diferentes propiedades de aspecto léxico. Vendler (1957) realizó una clasificación en cuatro grupos principales los verbos: estados, realizaciones (*accomplishments*),

actividades y logros (*achievements*). En la *Tabla 7* se exponen las características de cada uno de los verbos (Jaramillo *et al.*, 2005).

Tabla 7. Clasificación de los verbos

Tipo de verbo	Características	Ejemplos
Estado	<ul style="list-style-type: none"> - Evento que no ocurre sino que se da. - Situaciones no dinámicas. 	Saber, conocer, querer, amar, tener, ser, etc.
Actividad /proceso	<ul style="list-style-type: none"> - Evento que ocurre en el tiempo. - Situaciones dinámicas-durativas: sin referencia al punto final de la eventualidad. 	Andar, correr, caminar, leer, escribir, nevar, comer, reír, llorar, nadar, etc.
Realización	<ul style="list-style-type: none"> - Evento dinámico delimitado que progresa hacia un límite interno (tético). - Situaciones dinámico-durativas. 	Correr la maratón, escribir una carta, pintar, construir, etc.
Logro	<ul style="list-style-type: none"> - Evento dinámico delimitado, de duración muy breve, sin fases. - Culminando en punto. - Situaciones dinámicas puntuales sin duración. 	Alcanzar la cima, nacer, encontrar, reconocer, morir, llegar, florecer, etc.

De acuerdo con lo anterior, los eventos se asocian con verbos de tipo proceso en la categoría de ocurrencia en tiempo y no en situaciones de operaciones o actividad como lo detallan en la categoría 2 de situaciones dinámicas que se pueden ejecutar por un sujeto definido. En el complemento deben tener un experimentador, un beneficiario o, simplemente un tema, pero nunca un actor. Tal como se define en los patrones semánticos hay unas características especiales que permiten definir la forma de utilización del verbo para la definición de un evento. Por ejemplo “la campana suena” (para mayor claridad observar la sección siguiente), es una oración que no requiere un actor asociado y, por lo tanto, se puede precisar que en la forma que está escrita indicaría que es un evento.

3.1.2 Patrones semánticos

La valencia de un verbo se define como su capacidad para relacionarse con diferentes elementos de la oración, particularmente actantes (relacionados con los actores que lo ejecutan) o circunstantes (circunstancias ligadas con el verbo mismo). A nivel

semántico, uno de los patrones característicos de los verbos que representan eventos se relaciona con la valencia cero (no requieren actantes ni circunstantes) o uno (requieren sólo un circunstante). Algunos verbos de eventos también pueden presentar dos o más valencias.

- **Verbos sin valencias:** Los verbos que no presentan ninguna valencia son los verbos que sólo se conjugan en tercera persona y que no tienen sujeto. Algunos ejemplos son: *llover*, *granizar* y *nevar*. Por ejemplo el verbo *lloverse* define como el agua que cae a la tierra en forma de gotas; para este verbo sólo se considera esta descripción (Galicia, 2000).
- **Verbos con una valencia:** Algunos verbos tienen únicamente la valencia que corresponde al complemento, pero no tienen un sujeto asociado (Galicia, 2000). Por ejemplo, para el verbo *sonar*, se tiene la siguiente valencia: objeto X emite un sonido. $1 = X$; ¿qué suena? % una canción ~, % una campana ~. En la Tabla 8 se muestran verbos de valencia cero o uno y su uso.

Tabla 8. Definición de evento disparador tipo mensaje y señal

Eventos disparadores de tipo mensaje y señal	
La representación de eventos que tienen valencia cero o uno, se describe para representar los eventos disparadores de tipo mensaje y señal	
Llegar Surgir Venir Dormir Aparecer Sonar Estornudar	Tintinear Nacer Volar Permitir Pasar Morir.

- **Verbos con dos o más valencias:** Los verbos con dos o más valencias se pueden considerar de eventos siempre y cuando se conjuguen de manera impersonal (utilizando voz pasiva refleja, por ejemplo), de forma que se exprese una carencia del sujeto que realiza la acción. Estos verbos pueden ser *sermonovalentes*, que son los que sólo tienen sujeto: *dormir* (yo duermo). El

verbo *beber* es *bivalente* (yo bebo agua), pues tiene dos actantes. El verbo *dar* es *trivalente* (yo doy un regalo a Pedro). La mayor parte de los verbos usuales son trivalentes como *estar* o *hacer* (Galicia, 2000).

- **Complemento:** En la caracterización semántica de complementos para la representación semántica de eventos, el complemento debe cumplir alguno de los siguientes roles: *Beneficiario* es un participante en una predicación verbal de tipo acción o proceso que no sufre cambios físicos a lo largo de la acción. Por ejemplo “usuario llega”. *Experimentadores* el elemento que designa a un participante de la predicación verbal que no es un partícipe consciente o voluntario del estado o proceso que predica el verbo. Típicamente, los sujetos de las oraciones intransitivas que designan estado o actividad son experimentadores, Por ejemplo “animal vuela”. *Tema* es un participante en una predicación verbal de tipo acción o proceso que no sufre cambios físicos a lo largo de la acción Por ejemplo “propuesta aparece”. Por ejemplo la campana suena, La Tablas 9 posibilita la definición del evento disparador tipo temporizador y la Tabla 10 compendia la definición de los eventos disparadores.

Tabla 9. Definición de eventos disparador tipo temporizador

Tipo de evento	Restricción	Ejemplo
Temporizador	El verbo se restringe a aquél que se puede asociar con el tiempo, como: transcurrir, pasar y llegar. Además, el sintagma nominal debe pertenecer al dominio de las representaciones de tiempo	Para indicar una fecha específica como: lunes, noviembre, 12/05/2012, cinco días. Algunos ejemplos de este tipo de eventos son: “llega el lunes” y “transcurren cinco días”.

Tabla 10. Representación general de los eventos disparadores

Representación léxica y sintáctica del evento disparador
Evento disparador →(verbo con valencia 0 o 1) o (verbo con valencia superior conjugado en voz pasiva refleja) + complemento
Complemento → sintagma nominal

Para observar la aplicación de la estructura definida en la Figura 7, por ejemplo, el evento sería “propuesta aparece”, acogiendo las restricciones sintácticas y semánticas establecidas. En efecto, “aparece” es un verbo al que no se le asigna un sujeto y “propuesta” es un sintagma nominal con el rol semántico “tema”.

3.1.3 Patrones sintácticos

Se identifican patrones sintácticos para el evento de resultado, a partir de la definición léxica y semántica, dado que es posible tener verbos de operación ligados con el evento. Así, la representación del evento de resultado se compendia en la Tabla 11.

Tabla 11. Representación sintáctica del evento de resultado

Tipo de evento	Restricción	Ejemplos
Resultado	Evento de resultado → sintagma nominal + verbo en participio pasado	Comentario eliminado y etiqueta modificada

3.2 Grafo de Interacción de eventos para eventos disparadores y de resultado

Con el objetivo de solucionar los problemas de consistencia para los eventos disparador y de resultado en UNC-Method, se realizó una adaptación al grafo de interacción de eventos de Yuan y Memom (2010) para mostrar la relación, la secuencia y el orden de ocurrencia de los eventos. En este tipo de diagramas se usan eventos como acciones u operaciones que un usuario realiza en la interfaz gráfica de usuario GUI. Para modelar estos eventos se utilizan diferentes modelos de representación para la creación de casos de prueba. Entre esos modelos están: *Event Flow Graph*, EFG Memom (2001) y Memom *et al.* (2007), *Event Sequence Graph*, ESG (Belli *et al.*, 2005), *Event Interaction Semantic Graph* ESIG (Yuan & Memom, 2010a), y *Event Interaction Graph*, EIG (Yuan & Memom, 2010b). Dado que este último se utiliza para representar la interacción de los eventos en la GUI, se emplea su sintaxis y se modifica para representar la interacción de eventos disparadores y de resultado en

UNC-Method. Esta adaptación del EIG permite visualizar la relación, el orden de ocurrencia y la secuencia entre los eventos. Además, se elige la notación de los eventos en los esquemas preconceptuales, que representan de forma precisa los eventos disparadores y de resultado, de tal manera que cualquier tipo de persona los pueda interpretar (Zapata, 2012).

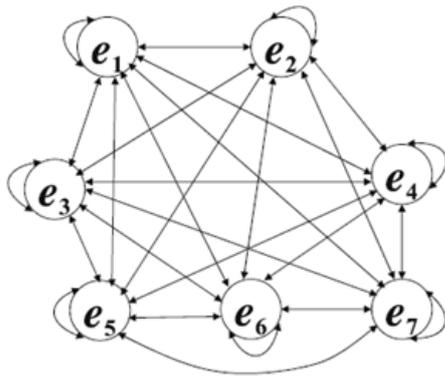


Figura 16. EFG

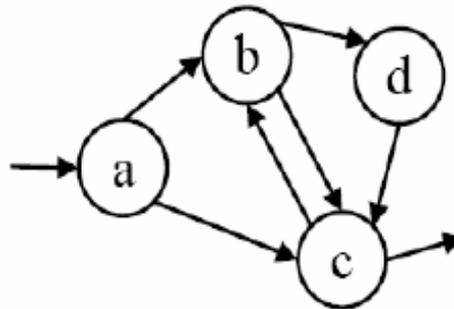


Figura 15. ESG

3.2.1 EIG. Grafo de interacción de eventos

El grafo de interacción de eventos es una herramienta que 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 (Watson & MaCabe, 1996). También, representan de manera gráfica los modelos de tiempo en donde los nodos que simbolizan los eventos se conectan por medio de 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 puede ser falsa o verdadera, además de un tiempo t que simboliza un retardo para la ejecución del evento (Urquía, 2003; Pan *et al.*, 2009). Los grafos de interacción de eventos representan la relación y ocurrencia de los eventos de la GUI para generar casos de prueba (Yuan & Memon, 2010c). El EIG es un refinamiento del EFG (Yuan & Memon, 2010b), al igual que el ESIG, Un EIG, además, representa todas las posibles secuencias de eventos que se pueden ejecutar en la interfaz gráfica de usuario

(Katamaya *et al.*, 1996), a diferencia de los demás modelos, que representan los eventos de naturaleza estructural de la GUI.

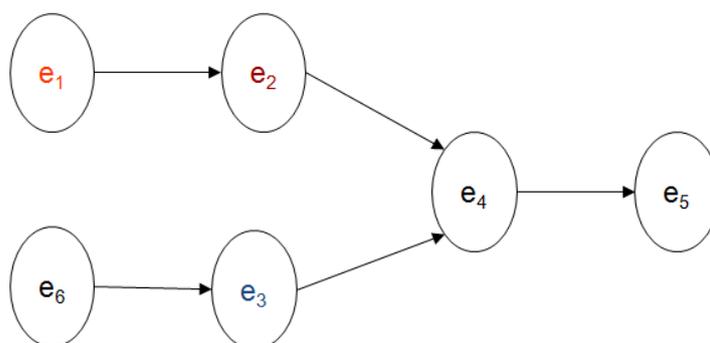


Figura 17. EIG

3.2.2 Representación de eventos disparadores y de resultado

Para la representación de eventos disparadores y de resultado se pretende utilizar la notación de eventos de los esquemas preconceptuales, donde se utilizan elementos de BPMN y se especifican de manera detallada los diferentes tipos de eventos, como se aprecia en las Figuras 18 y 19. Para el nuevo artefacto, se distinguen los eventos disparadores con un círculo rojo y los eventos de resultado con un círculo azul.

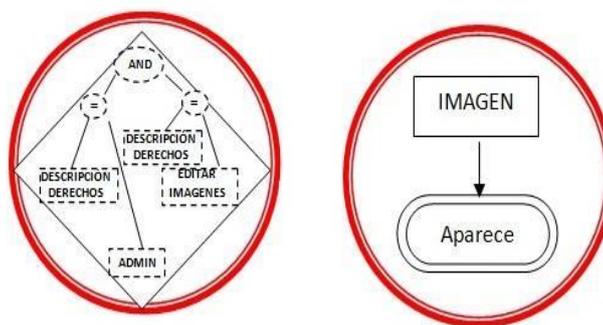


Figura 18. Notación para eventos disparadores



Figura 19. Notación para eventos de resultados

Una de las restricciones importantes dentro del EIG es que no se deberían conectar dos eventos de resultado de forma continua. Se puede unir un evento de resultado a un evento disparador significando el orden en la secuencia más no que proviene del evento de resultado.

3.2.3 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 los procesos se utiliza el joint/fork del diagrama de procesos (línea vertical) con su nombre respectivo que precede un proceso o un evento de resultado.

Se ejemplifica en la Figura 20 el grafo de interacción de eventos mediante el dominio de una red social ZenPhoto, donde se manejan álbumes, imágenes y etiquetas. Se pueden observar los eventos disparadores de color rojo: 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. Los eventos de resultado- álbum publicado - álbum editado- álbum eliminado - imagen subida- imagen editada - la imagen es etiquetada. Y los procesos crear álbum - publicar álbum- editar álbum - eliminar álbum- subir imagen - editar imagen- etiquetar imagen.

En este grafo de interacción de eventos se representan todas las posibles secuencias:

Secuencia 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.

Secuencia 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.

Secuencia 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.

Secuencia 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.

Secuencia 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.

El grado mediante las anteriores secuencias describe la relación entre los procesos y eventos y el orden de ocurrencia entre los eventos.

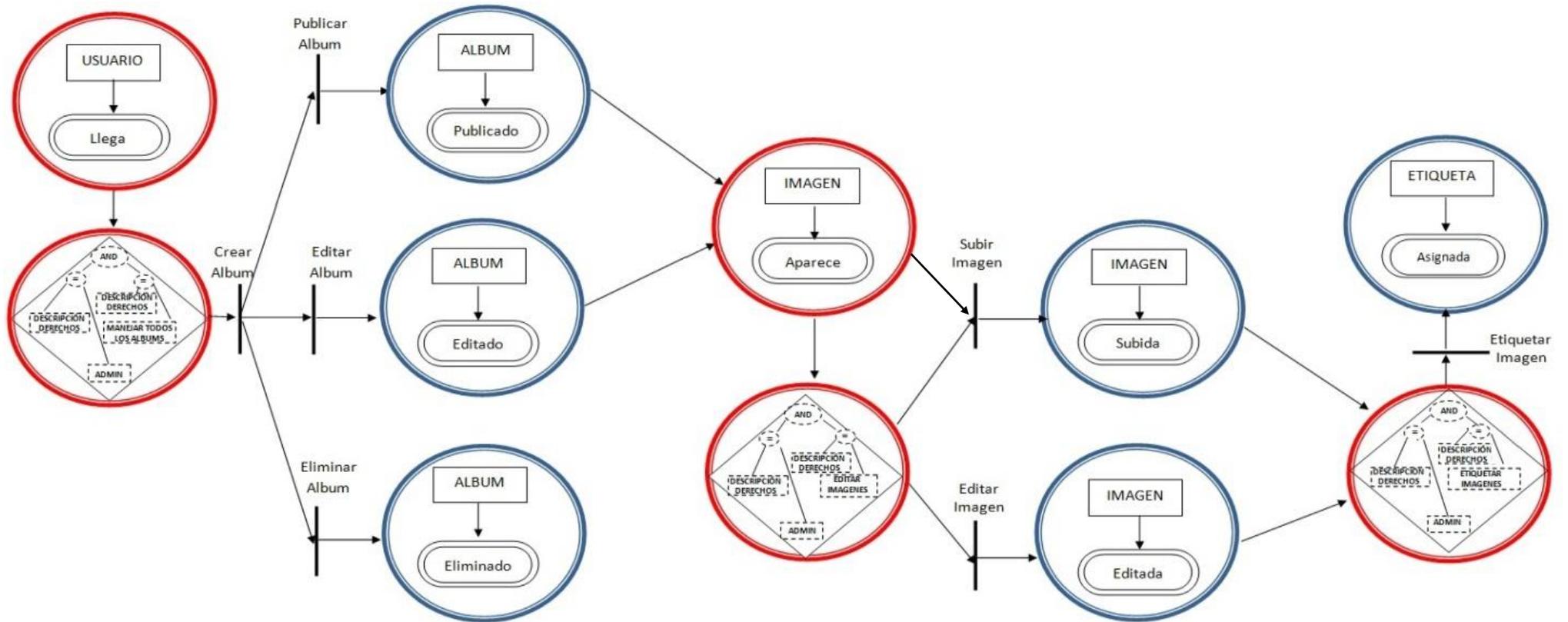


Figura 20. Grafo de interacción de eventos para los eventos disparadores y de resultado

3.3 Reglas de consistencia

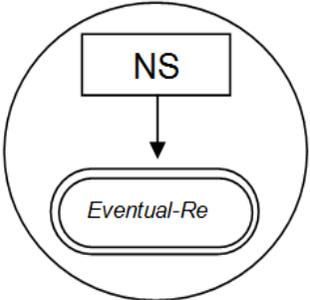
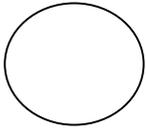
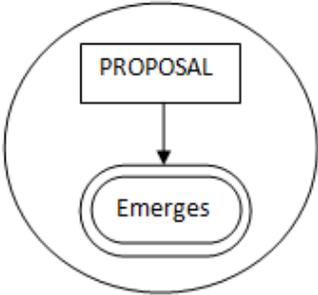
Se definen reglas de consistencia para los eventos disparadores y de resultado en los artefactos que los representan en UNC-Method, garantizando la consistencia que debe existir para la correcta definición del dominio del problema y la descripción la solución de un proyecto de software. Estas reglas se compendian en la Tabla 12 y se definen en inglés, pues es el lenguaje actual en el que se maneja el UNC-Method (Zapata, 2012). Si bien el diagrama de interacción de eventos será objeto de la próxima subsección, por efectos de plenitud se incluyen sus reglas de consistencia en la Tabla 12, previa a la definición de su aplicación en UNC-Method.

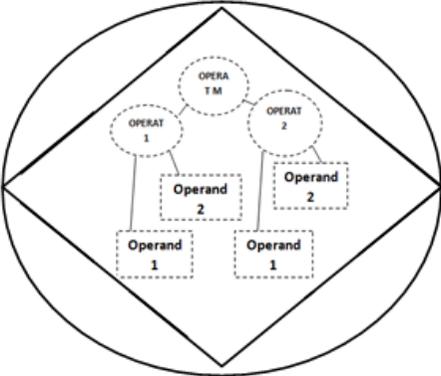
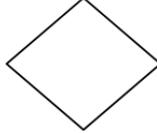
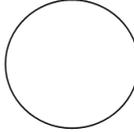
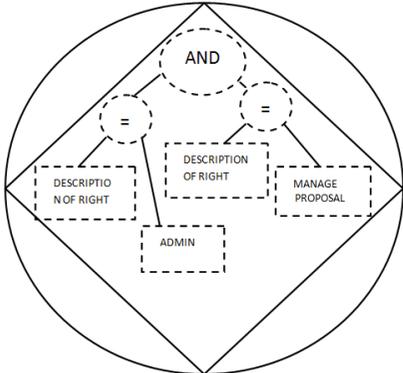
Tabla 12. Reglas de consistencia para los eventos en artefactos de UNC-Method

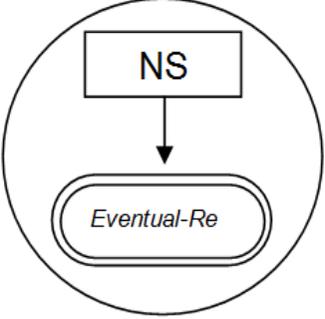
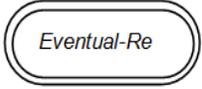
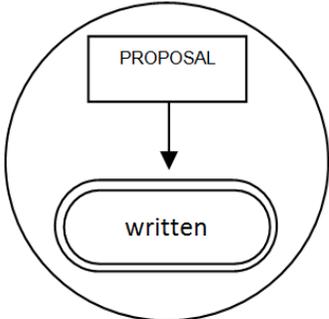
EVENT CONSISTENCY RULES				
Artifact	Description	Constraint	Definition	Example
1. Controlled dialogue	TriggerEvent			
	<p>CR1. <Dialogue >::= <Quest>+<Answ></p> <p>CR 1.1<Quest> ::=“What does the”+<person>+<“need in order to accomplish the”>+<Concept>+<Verb1>+<“?”></p> <p>CR 1.1.1<Concept>::=<NP></p> <p>CR 1.1.2<NP>::=<N> <P>+<N> <D>+<A *>+<N>+ <PP></p> <p><PP>::=<P>+<NP></p> <p>CR 1.2<Answ>::=<TriggeredAnsw> <ConditionalAnsw></p> <p>CR 1.2.1<TriggeredAnsw>::=“He just needs”+<person>+ <verb2>+<“of”>+<“a”> <“an”>+<concept></p>	<p>Verb1 {is conjugated in gerund, i.e. “painting”, “uploading”, “publishing” or in nominalized form, i.e. “creation”, “Edition” }</p> <p>D {is a determinant like “the”, “some”}</p> <p>A {one or more adjectives, i.e. big, brown, old }</p> <p>verb2 {is conjugated in gerund, i.e. “arising”, “arriving”, “coming”}</p>	<p>Answ: Answer</p> <p>Eventual-Re: Eventual relationship</p> <p>Quest: Question</p> <p>NP: Noun phrase</p> <p>N: Nouns</p> <p>P: Preposition</p> <p>Person: Actor</p> <p>PP: Prepositional Syntagm</p> <p>pp: Past participle</p> <p>D: Determiners</p> <p>A: Adjective</p> <p>Verb1, verb2: Verbs</p>	<p>Dialogue 1:</p> <p><i>Quest:</i> <i>What does the RESEARCHER need in order to accomplish the PROPOSAL WRITING?</i></p> <p><i>Answ:</i> <i>TriggeredAnsw: He just needs the EMERGENCE of a PROPOSAL</i></p> <p><i>Quest:</i> <i><What does the>+<RESEARCHER>+<need in order to accomplish the> + <PROPOSAL> +< WRITING> + <?></i></p> <p><i>Answ:</i> <i>TriggeredAnsw: <He just needs the> +<EMERGENCE>+<of>+<a><PROPOSAL></i></p>

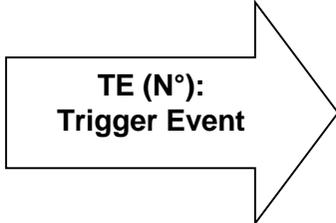
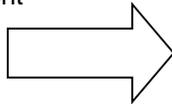
<h2 style="writing-mode: vertical-rl; transform: rotate(180deg);">1. Controlled dialogue</h2>	<p>CR 1.2.2 <ConditionalAnsw>::=<“He just needs to have”>+<Cond-Complete></p> <p><Cond-Complete>::=<Cond> + (<operat> + <Cond>)*</p> <p><Cond>::=<Operand1> + <operator> + <Operand2></p> <p><Operand1>::=<Concept></p> <p><Operator2>::=<“Concept”> <“Numeric Expression”></p>	<p>Cond-Complete: {Has one or more conditions. If it has more than one conditions, it needs a new operator}</p> <p>Operat: {could be either arithmetic, logic, or relational</p> <p>i.e. arithmetic: +,-, /,* Logic: AND, OR, relational: =, !=, <, >, <=, >=</p> <p>In this case, its name should be <i>operat1</i>, <i>operat2</i>, etc., and main operator would be <i>operatm</i> (can be a logic operator like AND and OR.)</p>	<p>Cond-Complete: several conditions.</p> <p>Cond: Condition</p> <p>Operat: Operator</p> <p>Operatorm: main operator</p>	<p>Dialogue 2:</p> <p>Quest: <i>What does the RESEARCHER need in order to accomplish the PROPOSAL EDITION and MANAGEMENT?</i></p> <p>Ans: <i>ConditionalAnsw: He just need to have description right =“Admin” AND description right =“manage proposals”</i></p> <p>Quest: <i><What does the>+ <RESEARCHER>+<need in order to accomplish the>+<PROPOSAL>+<EDITION and MANAGEMENT>?</i></p> <p>Ans: <i>ConditionalAnsw: He just need to have <description right> =<“Admin”>+<AND>+<description right =“manage proposals”></i></p> <p><i>Operat1>::= “=” <operand1>::=<description right> <operand2>:<“Admin”> <cond1>::=<description right> = <“Admin”></i></p> <p><i><Operat2>: = <operand1>::=<description right> <operand2>::=<“manage proposals”> <cond2>::=<description right> = <“manage proposals”></i></p> <p><i><Cond-Complete>::= Operatorm::=AND<description right> = <“Admin”> + <AND> + <description right> =<“manage proposals”></i></p>
---	--	--	--	--

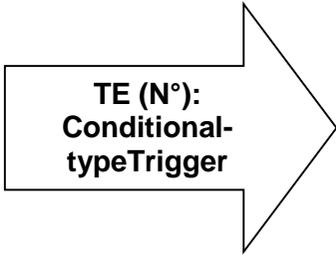
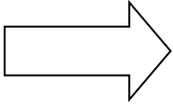
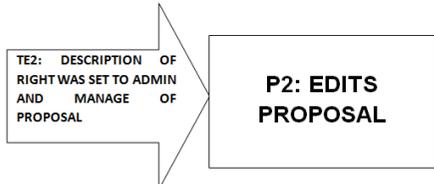
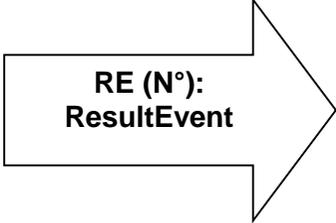
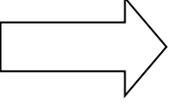
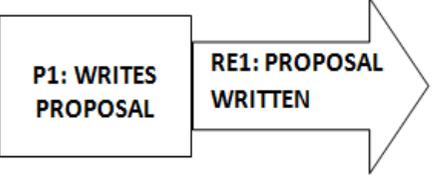
2. Elicitation cards		Trigger Event																									
		<p>CR 2. Cards</p> <p>Replace the information created in controlled dialogue to trigger event with condition in <verb1>, <person>, <concept> and <Cond-Complete></p> <table border="1" data-bbox="398 644 730 967"> <tr><td colspan="2">FUNCTION</td></tr> <tr><td colspan="2"><verb1></td></tr> <tr><td>ACTOR</td><td><person></td></tr> <tr><td>OBJECT</td><td><concept></td></tr> <tr><td colspan="2">CONSTRAINT</td></tr> <tr><td colspan="2"><Cond-Complete></td></tr> </table>	FUNCTION		<verb1>		ACTOR	<person>	OBJECT	<concept>	CONSTRAINT		<Cond-Complete>		<p>Verb:</p> <p>{should be in present time and third person}</p> <p>In the constraint:</p> <p>{The Cond-Complete has one or more conditions.</p> <p>If it has more than one condition, it needs a new operator}</p>	<p>Function: it is the verb 1 of the trigger event used in question of the controlled dialogue.</p> <p>Actor: He/she interacts with the object and performs the function.</p> <p>Object: is the concept. The actor performs the function on it.</p> <p>Related rules: CR 1.1.1 and CR 1.1.2 of the controlled dialogue.</p> <p>Constraint: is the actor need of the controlled dialogue.</p> <p>Related rule: CR 1.2.2 in the specification of Cond-Complete in the controlled dialogue.</p>	<p><i>See example of controlled dialogue, the question of dialogue 2.</i></p> <p><i>Function: Edits(3rd person)</i></p> <p><i>Actor: Researcher</i></p> <p><i>Object: Proposal</i></p> <p><i>Constraint: Description of right="Admin" and description of right ="manage proposals"</i></p> <table border="1" data-bbox="1559 860 1982 1115"> <tr><td colspan="2">FUNCTION</td></tr> <tr><td colspan="2">Edits</td></tr> <tr><td>ACTOR</td><td>Researcher</td></tr> <tr><td>OBJECT</td><td>Proposal</td></tr> <tr><td colspan="2">CONSTRAINT</td></tr> <tr><td colspan="2">Description of right="Admin" and description of right ="manage proposals"</td></tr> </table>	FUNCTION		Edits		ACTOR	Researcher	OBJECT	Proposal	CONSTRAINT	
FUNCTION																											
<verb1>																											
ACTOR	<person>																										
OBJECT	<concept>																										
CONSTRAINT																											
<Cond-Complete>																											
FUNCTION																											
Edits																											
ACTOR	Researcher																										
OBJECT	Proposal																										
CONSTRAINT																											
Description of right="Admin" and description of right ="manage proposals"																											

<p style="writing-mode: vertical-rl; transform: rotate(180deg);">3. Pre-conceptual schema</p>	<p>CR 3.Trigger event → <NS> + <Eventual-Re></p> <p>Replace consistently from rules created in controlled dialogue for trigger events on NS and Eventual-Re.</p> 	<p>Verb:</p> <p>{Should be conjugated in simple present</p> <p>It can have either valence 0 or 1 or more.</p> <p>When valence is 0: {the verb does not have an actor i.e. “rain”, “arrive”, “come”, “emerge”, “fly”,</p> <p>When valence is 1, the verb has an actor but does not have more information, i.e. “John looks”, “James buys”.</p> <p>When valence is more than 1 the phrase has more information and should be conjugated in passive voice. i.e. “Mateo has rented his house”. In this case, it should have 2 valences.}</p>	<p>Connection: The line is from NS to Eventual-Re.</p> <p style="text-align: center;">→</p> <p>NS=concept Related rules: CR1.1.1 and CR 1.1.2 of the controlled dialogue.</p>  <p>Eventual-Re: eventual relationships for trigger event should be the verb 2 in stakeholder answer of the controlled dialogue. Related rule: CR 1.2.1 of the controlled dialogue.</p>  <p>Event:</p> 	<p>See the example of the controlled dialogue, the stakeholder answer of dialogue 1 and the elicitation card example.</p> <p>NS: Proposal</p> <p>Verb 2: in controlled dialogue, it is <i>Emerging</i>.</p> <p>In pre-conceptual schema is <i>Emerges</i>.</p> 
--	---	--	--	--

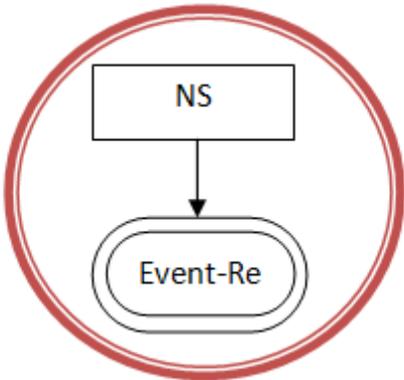
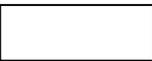
<p style="writing-mode: vertical-rl; transform: rotate(180deg);">3. Pre-conceptual schema</p>	<p>CR 3.1 Conditional-type Trigger Event → <Cond-Complete></p> <p>Replace with the information from <Cond-Complete> in the controlled dialogue and the elicitation card for the conditional-type trigger event on operat1, operand1 and operand2 of operat1. Operat2, operand1 and operand2.</p>  <p>The trigger event should be represented by a circle and it can have several conditions.</p>	<p>Conditional-type trigger event</p> <p>{Has one or more conditions. If it has more than one condition, it needs a new operator. Related rule: CR 1.2.2 in the specification of <i>Cond-Complete</i> in the controlled dialogue.}</p> <p>Operat: {it can have one or more operators. It could be either arithmetic, logic or relational. i.e. arithmetic: +,-,/,* logic: AND, OR, relational: =, !=, <,>,<=,>=}</p> <p>Operat M: {the main operator of conditions should be united either with an arithmetic or a relational operator}</p> <p>Operand1: Concept Operat2: {Could be a concept or a numeric expression}</p>	<p>Operat: operator</p>  <p>Operand: specification</p>  <p>Cond: conditional</p>  <p>Event</p>  <p>Operator M: main operator</p>	<p>See the example of the controlled dialogue, the stakeholder answer of the dialogue 2, and the constraint of the elicitation card.</p> <p>Cond1: Operat1: = Operand1: Description of right Operand2: Admin Cond2: Operat2: = Operand1: Description of right Operand2: Manage proposal</p> <p>Cond-Complete Operatm: AND Cond1 AND Cond2</p> <p>See the rule CR 1.2.2 for the specification of <i>Cond-Complete</i> in the controlled dialogue.</p> 
--	---	---	--	--

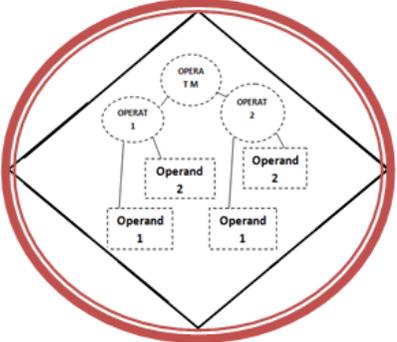
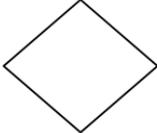
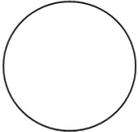
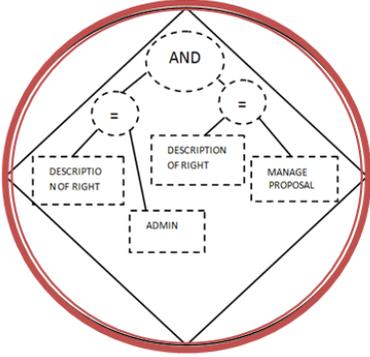
<p style="writing-mode: vertical-rl; transform: rotate(180deg);">3. Pre-conceptual Schema</p>	<p>CR 4. Result Event → <NS> + <Eventual-Re></p> <p>Replace the information from the controlled dialogue on NS and Eventual-Re.</p>  <p>The result event should be represented by a circle and it can have 3 elements: NS, Connection, and Eventual-Re.</p>	<p>Result event: { NS with the verb }</p> <p>Verb: {The verb should be conjugated in past participle, i.e. "painted", "created", "edited", "published", "done", "gone". }</p>	<p>Connection: The line goes from NS to Eventual-Re.</p> <p style="text-align: center;">→</p> <p>NS=concept Related rules: CR 1.1.1 and CR 1.1.2 of the controlled dialogue.</p> <p>Pp: past participle NS: noun phrase NS=concept</p>  <p>Eventual-Re: should be the verb 1 of the analyst question of the controlled dialogue. Related rule: CR 1.1 of the controlled dialogue.</p> 	<p>See example of controlled dialogue, the analyst's question of dialogue 1.</p> <p><i>Result Event:</i> NS: PROPOSAL Verb1: in controlled dialogue is Writing. In pre-conceptual schema, it is Written</p> 
--	---	---	---	---

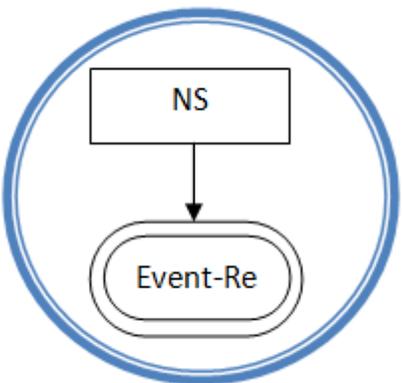
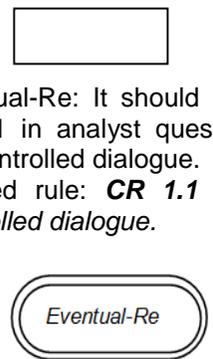
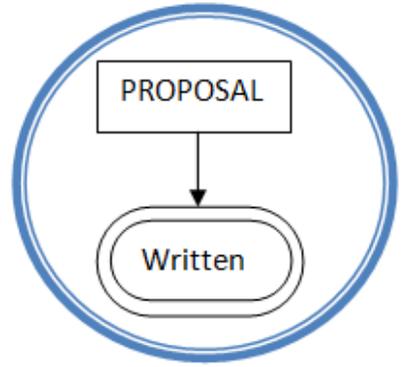
4. Process Diagram		Trigger Event		
		<p>CR5. TE(N°): Trigger Event</p> <p>Replace from pre-conceptual schema CR 3. Trigger event on TE (N°): Trigger Event</p> 	<p>Trigger Event: {it is always before a process, will have numbers according to the quantity.}</p> <p>Verb: {Should be in present in simple form</p> <p>This can be with valence 0 or 1 or more. When valence is 0: {the verb does not have an actor i.e. "rain", "arrive", "come", "emerge", "fly",</p> <p>When valence is 1, the verb has an actor, but does not have more information, i.e. "John looks", "James buys".</p> <p>When valence is more than 1, the phrase has more information, it should be in passive voice, i.e. "My bike was" in this case should have 2 valences.}</p>	<p>TE: trigger event</p> <p>Event</p>  <p>Process</p>  <p>N°: Code of trigger event</p>

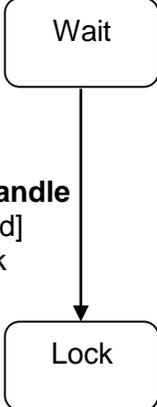
<h3>4. Process Diagram</h3>	<p>CR6. TE(N°): <operand1> was set to <operand2_c1><operatM><operand2_c2></p> <p>Replace from the information of the pre-conceptual schema rule CR 3.1 Conditional-type Trigger</p> <div style="text-align: center;">  </div>	<p>Trigger Event</p> <p>conditional:</p> <p>{it always goes before a process and it will have numbers according to the quantity}</p> <p>Operand1: {is written just one time if it is same in the different conditions}</p>	<p>Event</p> <div style="text-align: center;">  </div> <p>Process</p> <div style="text-align: center;">  </div> <p>Operand2_c1: operand of condition1 Operand2_c2: operand of condition2 operatM: main operator, is logic.</p> <p>N°: Code of trigger event</p>	<p>See the example of the controlled dialogue, the stakeholder answer of the dialogue 2, the constraint of elicitation card, and the trigger event conditional of the pre-conceptual schema.</p> <p>TE2: Description of right was set to admin and manage of proposal</p> <div style="text-align: center;">  </div>
	<p>CR7. RE(N°): result event</p> <p>Replace from the information of the pre-conceptual schema rule CR4.</p> <div style="text-align: center;">  </div>	<p>Result Event:</p> <p>{it goes after a process and it will have numbers according to the quantity}</p>	<p>Event</p> <div style="text-align: center;">  </div> <p>Process</p> <div style="text-align: center;">  </div> <p>N°: Code of result event</p>	<p>See the example of the controlled dialogue, the analyst question of the dialogue 1.</p> <p>RE1: Proposal written While it was gerund in the controlled dialogue, here it should be conjugated in past participle.</p> <div style="text-align: center;">  </div>

5. Process diagram explanatory table	<p>CR7. Relation of process with events</p> <p>Define P(N°), TE(N°), RE(N°) from the process diagram rules <i>CR5, CR6 and CR7</i>.</p> <table border="1" data-bbox="327 539 786 695"> <thead> <tr> <th>Name</th> <th>Event</th> </tr> </thead> <tbody> <tr> <td>P(N°)</td> <td>TE(N°)</td> </tr> <tr> <td></td> <td>RE(N°)</td> </tr> <tr> <td></td> <td></td> </tr> </tbody> </table>	Name	Event	P(N°)	TE(N°)		RE(N°)			<p>Event { All events related to process should be in Event (square part) }</p> <p>It should have the same code of the process diagram</p>	<p>TE: Trigger Event RE: Result Event N°: Code of event or process</p>	<p><i>See examples of the process diagram</i></p> <table border="1" data-bbox="1541 384 2000 635"> <thead> <tr> <th>Name</th> <th>Event</th> </tr> </thead> <tbody> <tr> <td>P1. Writes proposal</td> <td>TE1 Proposal emerges</td> </tr> <tr> <td></td> <td>RE1. Proposal written</td> </tr> <tr> <td></td> <td></td> </tr> </tbody> </table> <table border="1" data-bbox="1541 699 2000 954"> <thead> <tr> <th>Name</th> <th>Event</th> </tr> </thead> <tbody> <tr> <td>P2. Edits proposal</td> <td>TE2: Description of right was set to admin and manage of proposal</td> </tr> <tr> <td></td> <td></td> </tr> </tbody> </table>	Name	Event	P1. Writes proposal	TE1 Proposal emerges		RE1. Proposal written			Name	Event	P2. Edits proposal	TE2: Description of right was set to admin and manage of proposal		
Name	Event																									
P(N°)	TE(N°)																									
	RE(N°)																									
Name	Event																									
P1. Writes proposal	TE1 Proposal emerges																									
	RE1. Proposal written																									
Name	Event																									
P2. Edits proposal	TE2: Description of right was set to admin and manage of proposal																									

6. Event interaction graph	Trigger Event		
	<p>CR8. By using CR 3: <i>Triggerevent</i> → <NS> + <Eventual-Re></p> <p>Replace with the information of the consistency rules created in the controlled dialogue, the elicitation cards, the pre-conceptual schema, and process diagram in NS and Event-Re</p> <div style="text-align: center;">  </div> <p>The trigger event should be represented by a circle and it can have 3 elements: NS, Connection and Eventual-Re.</p>	<p>Verb:</p> <p>{Should be conjugated in simple present</p> <p>It can have valence 0 or 1 or more.</p> <p>When valence is 0: {the verb does not have an actor, e. g. “rain”, “arrive”, “come”, “emerge”, “fly”.</p> <p>When valence is 1, the verb has an actor, but does not have more information, i.e. “John looks”, “James buys”.</p> <p>When valence is more than 1, the phrase has more information and it should be conjugated in passive voice, e.g. “My bike was stolen”. In this case, it should have 2 valences.}</p>	<p>Connection: The line goes from one event to another.</p> <div style="text-align: center;">  </div> <p>NS=concept Related rules: CR 1.1.1 and CR 1.1.2 of the controlled dialogue.</p> <p>NS: Noun Phrase</p> <p>NS=concept</p> <div style="text-align: center;">  </div> <p>Eventual-Re: the result event has the verb 1 of the analyst question of the controlled dialogue. Related rule: CR 1.1 of the controlled dialogue.</p> <div style="text-align: center;">  </div> <p>Joint/fork: represents the process.</p> <div style="text-align: center;">  </div>

<h2 style="writing-mode: vertical-rl; transform: rotate(180deg);">6. Event interaction graph</h2>	<p>CR9. By using CR 3.1 Conditional-type Trigger: Event → <Cond-Complete></p> <p>Replace with the information of the consistency rules created for the controlled dialogue, the elicitation cards, the pre-conceptual schema, and the process diagram.</p>  <p>The trigger event should be represented by a circle and it can have several conditions.</p>	<p>Conditional-type Trigger Event {Has 1 or more conditions. If it has more than one condition, it will have a new operator Related rule: CR 1.2.2 in the specification of <i>Cond-Complete of the controlled dialogue.</i>}</p> <p>Operat: {Can be 1 or more operators.</p> <p>could be arithmetic, logic or relational</p> <p>i.e. arithmetic: +, -, /, * logic: AND, OR, relational: =, !=, <, >, <=, >=}</p> <p>Operat M: {the main operator of conditions should be united with either an arithmetic or a relational operator}</p> <p>Operand1: Concept</p> <p>Operat2: {Could be either a Concept or a Numeric Expression}</p>	<p>Operat: operator</p>  <p>Operand: Specification</p>  <p>Cond: Conditional</p>  <p>Event</p>  <p>Operator M: main operator</p> <p>Joint/fork: it represents the process.</p> 	<p>See the example of the controlled dialogue, the stakeholder answer of the dialogue 2, and the constraint of the elicitation card.</p> <p>Cond1: Operat1: = Operand1: Description of right Operand2: Admin</p> <p>Cond2: Operat2: = Operand1: Description of right Operand2: Manage proposal</p> <p>Cond-Complete Operatm: AND Cond1 AND Cond2</p> <p>See rules CR 1.2.2 in the specification of <i>Cond-Complete</i> in the controlled dialogue.</p> 

<p>6. Event interaction graph</p>	<p>CR 9. Result Even → <NS> + <Eventual-Re></p> <p>Replace with the information of the consistency rules created for the controlled dialogue, the elicitation cards, the pre-conceptual schema and the process diagram in NS and Event-Re.</p>  <p>The result event should be represented by a circle, and it can have 3 elements: NS, Connection and Eventual-Re</p>	<p>Result event: { NS with the verb }</p> <p>Verb: {The verb should be conjugated in past participle, i.e. "painted", "created", "edited", "published", "done", "gone". }</p>	<p>Connection: The line goes from NS to Eventual-Re.</p> <p style="text-align: center;">→</p> <p>NS=concept Related rules CR 1.1.1 and CR 1.1.2 of the controlled dialogue.</p> <p>Pp: past participle NS: Noun Phrase NS=concept</p>  <p>Eventual-Re: It should be the verb 1 in analyst question of the controlled dialogue. Related rule: CR 1.1 of the controlled dialogue.</p> <p>Joint/fork: it represents the process.</p> <p style="text-align: center;"> </p>	<p>See the example of the controlled dialogue and the analyst question of dialogue 1.</p> <p>Result Event: NS: PROPOSAL</p> <p>Verb1: in the controlled dialogue, it is Writing.</p> <p>In the pre-conceptual schema, it is Written</p> 
--	---	---	---	--

7.State Machine diagram	<p>CR 9.States</p> <p>See the trigger event in rules CR2, CR3, and CR5.</p> <p style="text-align: center;">Trigger event [Guard] Activity</p> <p style="text-align: center;">→</p> <p>Commonly, the state machine diagram present the trigger event like result event.</p>	<p>transition</p> <pre>{ It should have 3 elements, something are 3 or 2 or 1 element. }</pre> <p>Trigger event should be presented like the rules of the previous artifacts.</p>	<p>state:</p>  <p>Transition:</p> <p style="text-align: center;">→</p> <p>Guard: condition</p> <p>Activity: action</p> <p>start state:</p>  <p>end state:</p> 	<p>Example:</p>  <pre> graph TD Wait([Wait]) -- "[door closed] Reveal lock" --> Lock([Lock]) </pre> <p>Remove candle [door closed] Reveal lock</p>
--------------------------------	--	---	--	---

4. Prototipo

En términos generales, el prototipo se define como apoyo para complementar las reglas de consistencia establecidas para los eventos en UNC-Method. Su objetivo principal es identificar eventos en el lenguaje natural a partir del diálogo controlado entre analista e interesado.

Algunos de las herramientas que se utilizan para el análisis morfológico de palabras o frases son las siguientes:

Analizadores morfológico de Verbos, AMV:

AMV, Es una solución para la conjugación y lematización de verbos en español, es una aplicación que reconoce la estructura interna de la mayoría de verbos en español. Donde se presenta la diferencia de verbos según los tipos especificados por Vlender. (Zapata, 2010).

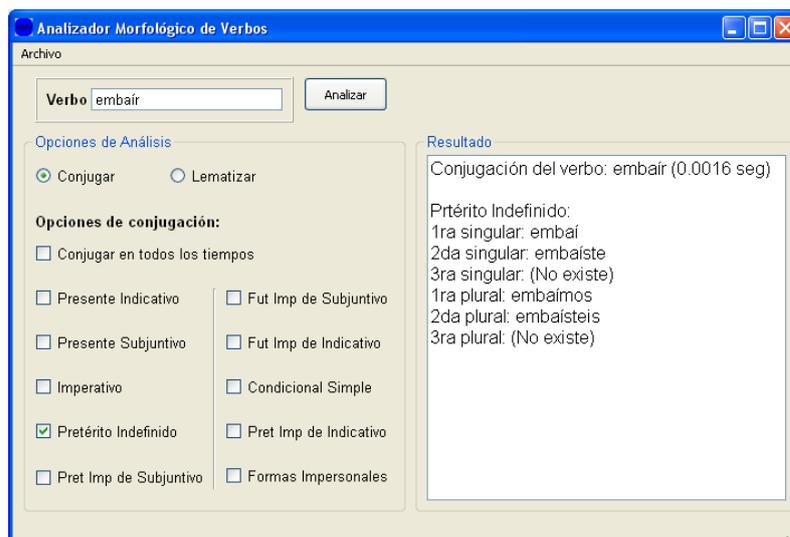


Figura 21. AMV

Diccionarios FrameNet

El proyecto FrameNet es creado por Fillmore como una base de datos léxica y un diccionario de más de 10.000 palabras, que muestravel significado y da ejemplos de su uso.

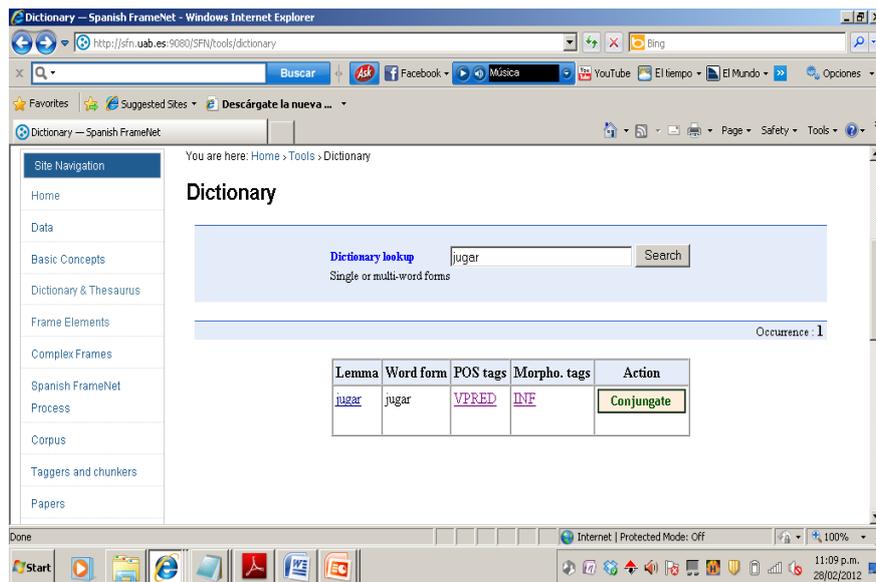


Figura 22. Diccionario morfológico

Sistema de Análisis y Generación de la Morfología, AGME

El Sistema de Análisis y Generación de la Morfología del Español, es un modelo que consiste en un conjunto de reglas para obtener todas las raíces de una forma de palabra para cada lexema y realiza su almacenamiento en el diccionario. (Gelback, 2009).

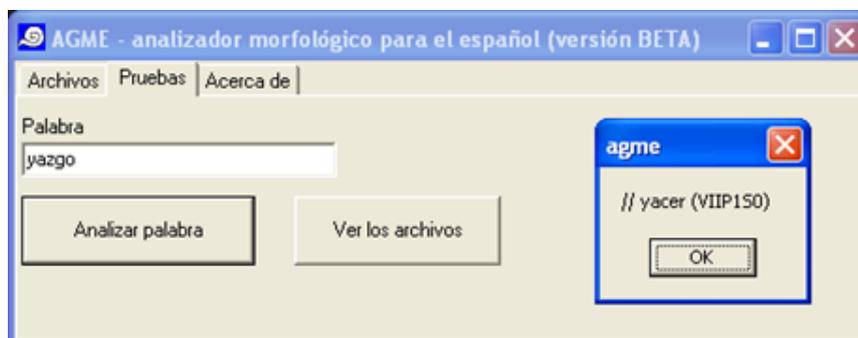


Figura 23. Analizador morfológico

4.1 Interfaz gráfica

La interfaz gráfica se divide en tres partes: diálogo controlado, palabras importantes y artefactos como se observa en la *Figura 24*.

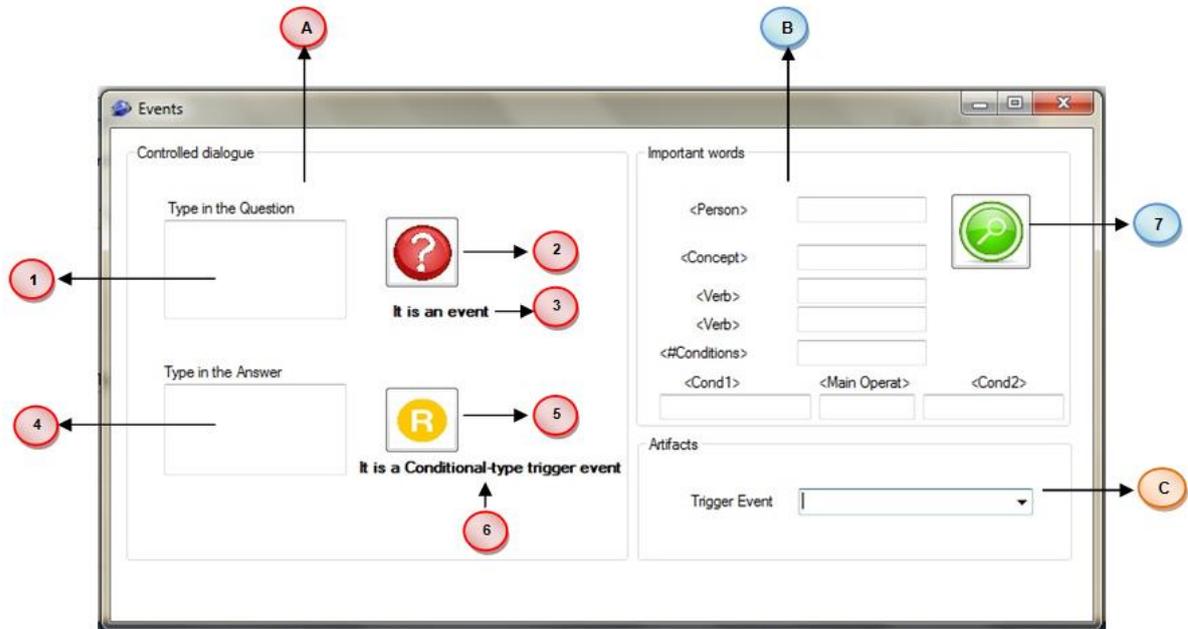


Figura 24. Interfaz gráfica del prototipo

- A** Diálogo controlado: El sistema pretende analizar cuáles preguntas y respuestas en la entrevista entre analista e interesado se refieren a eventos. Esta parte contiene:
 - 1** Campo de texto, para llenar la pregunta del analista.
 - 2** Botón de envío de pregunta (?), el cual determina si es un evento. En caso de serlo, activa la siguiente parte de respuesta del interesado.
 - 3** Mensaje de respuesta y verificación, donde el sistema muestra si es un evento.
 - 4** Campo de texto para llenar la respuesta del interesado.
 - 5** Botón de envío de respuesta @, para identificar el tipo de evento.
 - 6** Mensaje de respuesta y verificación, donde el sistema muestra el tipo de evento activando los campos que puede representar.
- B** Palabras importantes: Son palabras extraídas del dialogo controlado que se requieren en los diferentes artefactos que representan eventos. Es importante

aclarar que el prototipo requiere las reglas de consistencia que se definieron. Consta de varios campos como: <Person>, <Concept>, <Verb> (el primer verbo se trae de la pregunta el segundo verbo de la respuesta) para eventos sin condición y <#conditions>, <cond1>, <cond2>, <mainoperat> (en la reglas =<operatm>) para eventos condicionales. La extracción de estos campos logra facilitar la representación de eventos en los artefactos porque es la información que se requiere y la que se utiliza en las reglas de consistencia.

Botón de búsqueda, permite traer los campos que se necesitan para representar los eventos.

- c Muestra cuales artefactos de UNC-Method pueden representar eventos.

4.2 Funcionamiento

Mediante el primer campo de texto se introduce la pregunta del analista y mediante el botón se envía al sistema. Al verificar la información, se arroja una respuesta indicando que se trata de un evento o no. En la Figura 25 se puede observar que, al no tratarse de un evento, el sistema responde que no se encuentra un evento.

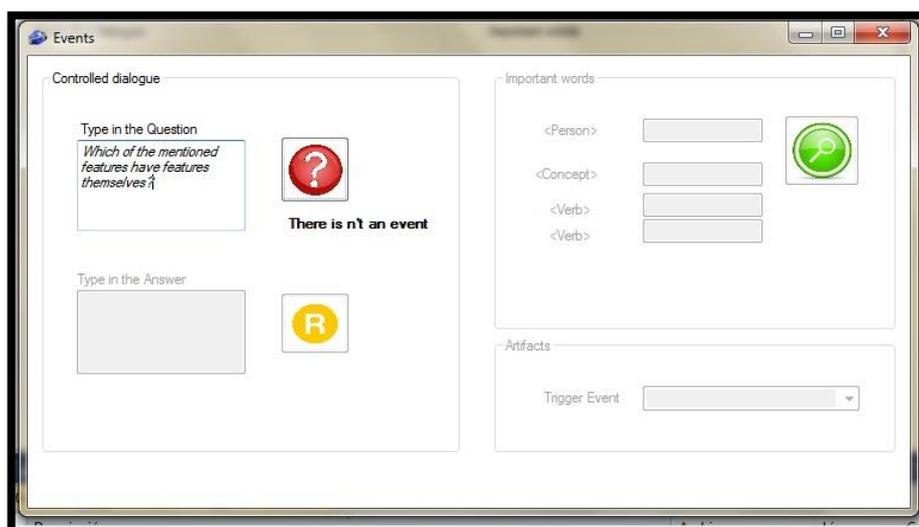


Figura 25. Pregunta en el dialogo controlado sin evento.

Cuando hay correspondencia con un evento, el sistema da la respuesta de que es un evento y activa la siguiente parte de la respuesta del interesado. Se llena el campo con la información de la respuesta y se oprime el botón R, para enviarla como se evidencia en la Figura 26.

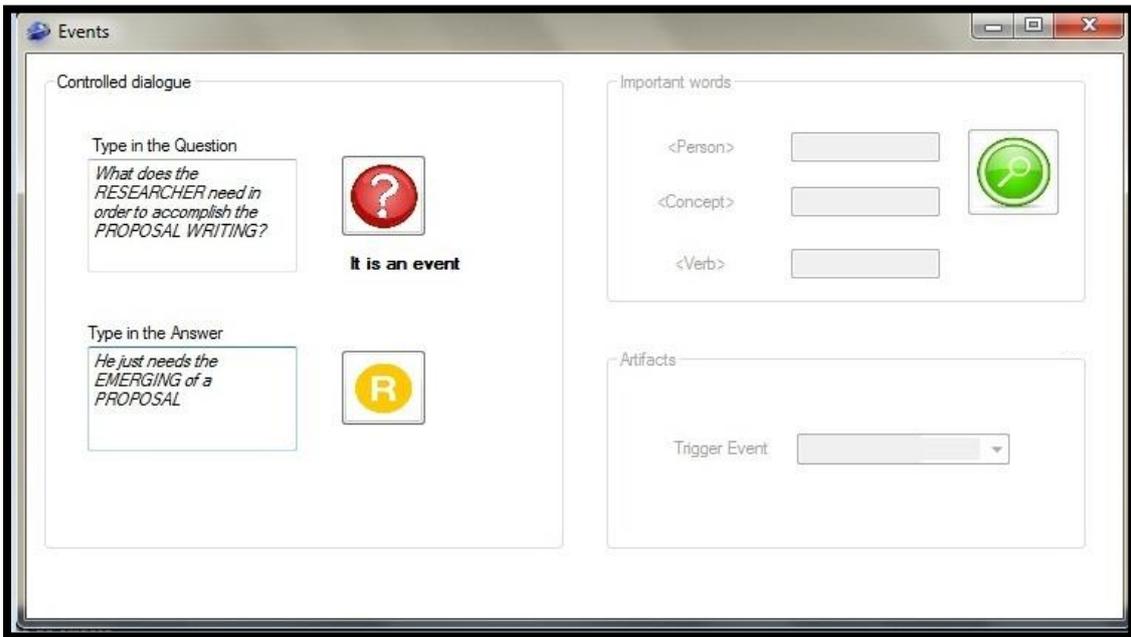


Figura 26. Pregunta en el dialogo controlado con evento.



Figura 27. Evento disparador en el dialogo controlado.

Como se observa en las Figuras 26 y 27, al presionar el botón R, el sistema en la respuesta determina el tipo de evento y activa las otras dos partes: palabras importantes y artefactos.

Al presionar el botón de búsqueda el sistema trae las palabras que se requieren en los artefactos posteriores que se van a realizar.

En el caso de la Figura 26, se describe un evento disparador y en la Figura 28 un evento disparador de tipo condicional, donde se puede observar que el sistema trae también la información de la condición.

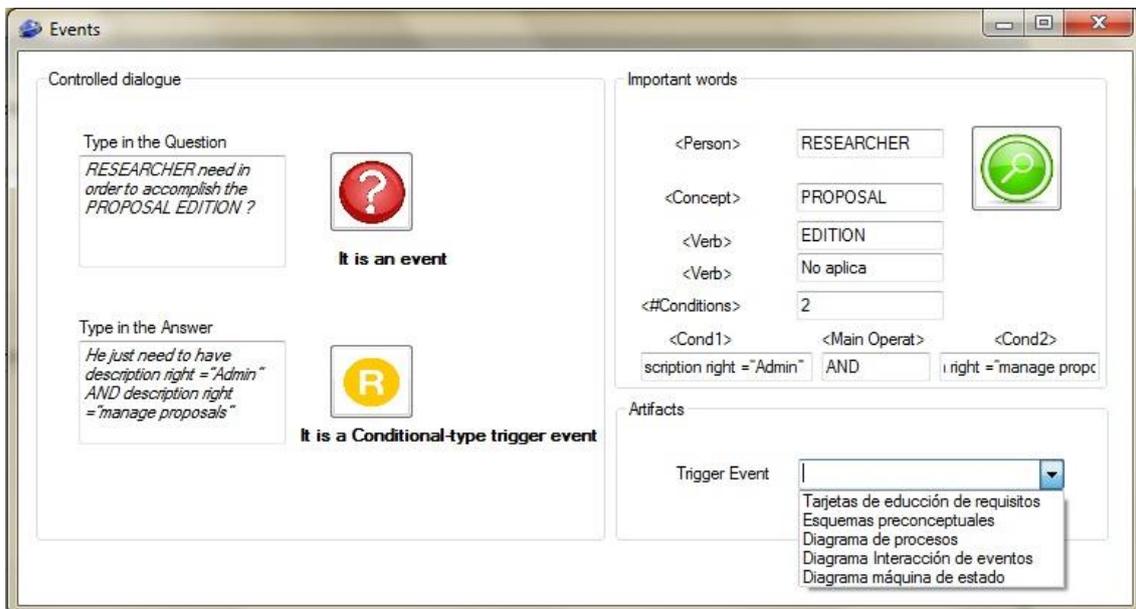


Figura 28. Evento Condicional en el dialogo controlado.

5. Validación del mecanismo

Se pretende en este Capítulo validar el mecanismo propuesto con el uso de un caso de estudio que ejemplifique la representación de los eventos disparadores y de resultado en los artefactos de UNC-Method. El caso de estudio evaluado pertenece a los estudiantes Carlos Andrés Toro, Oscar Eduardo Molano y Ricardo Mesa Foronda del curso de ingeniería de requisitos de la Universidad Nacional de Colombia, durante el semestre 01/2013. En este caso de estudio se pretende evaluar el cumplimiento actual de las reglas de consistencia definidas para los eventos. Es de notar que los estudiantes no son hablantes nativos en inglés y que los artefactos se presentan de la manera más fidedigna posible, sin hacer correcciones al idioma, pues el objetivo es mostrar el manejo de los eventos y no realizar correcciones lingüísticas.

5.1 Diálogo controlado

Analyst: Good morning. With this interview, we aim to clarify the information concerning the problem domain in which we will work. Please answer the questions in the clearest way possible.

Stakeholder: O.k. Let's start.

Analyst: What is the name of this organization?

Stakeholder: HMO Research Network.

Analyst: What is the general goal of the organization?

Stakeholder: The HMORN is not a legal or corporate entity, however, functions as an institution, which objective is, independently of organizations, sharing resources to reduce the burden on investigators from one HMORN member who propose to conduct research in multiple HMORN member institutions.

Analyst: Would you please mention some characteristics of HMORN?

Stakeholder: Yes, it has institutions, and every institution has an institution name, also, for each institution, there are one or more researches.

Analyst: Please mention some characteristics of these institutions?

Stakeholder: Yes, In addition every institution has Investigators and IRBs (Institutional Review Board) and those IRBs has requirements to approve a research.

Analyst: Which of the mentioned features have features themselves?

Stakeholder: Yes, every IRB can be categorized as Lead IRB, Reviewing IRB or Ceding IRB, also investigators has research burden.

Analyst: Please mention some features of the research?

Stakeholder: A research has title, number, protocol, information, review authority, document and regulation.

Analyst: Which of the mentioned features have features themselves?

Stakeholder: The documentation has type (HIPAA privacy rule determination, data use agreement, business associate agreement, approve authorization form, waiver authorization form, alteration authorization form, HMORN Inter-Institutional Research Application, which has IRB Application Form and Cover Sheet), approval status, approval date, creation date; protocol has protocol medication status, review authority has review authority status.

Analyst: Please, list the internal/external actors linked to the activities of your organization.

Stakeholder: We have Administrator and Investigator.

Analyst: Please, list the roles associated to the investigator?

Stakeholder: The investigator can be Lead PI or Non Lead PI

Analyst: Please, list the roles associated to the administrator?

Stakeholder: The administrator can be Lead IRB Administrator, Ceding IRB Administrator or Reviewing IRB Administrator.

Analyst: Would you please mention some characteristics of investigator?

Stakeholder: They have Name, research burden, Institution Name, Phone, Address, Fax and Email.

Analyst: What does the Ceding IRB Administrator need in order to accomplish the 'Documents the Review Authority Status'?

Stakeholder: He just needs to have Review Authority Status= 'Ceded'

Analyst: What does the Reviewing IRB Administrator need in order to accomplish the 'Notifies of Review Authority Status'?

Stakeholder: He just need to have Review Authority Status= 'Not Ceded'

Analyst: What does the user need in order to accomplish the 'Modifies Protocol'?

Stakeholder: He just need to have Protocol Modification Status='Approved'

Analyst: What does the user need in order to accomplish the 'Notifies Status'?

Stakeholder: He just need to have Status= 'Complete'

Analyst: What are the goals and problems associated with the 'research burden'?

Stakeholder: The goal is 'Reducing research burden'. The problem is 'The investigators have a heavy research burden'

Analyst: What are the goals and problems associated with the 'Investigator'

Stakeholder: The goal is 'Increasing Investigator'. The problem is there are few Investigators.

Analyst: What are the goals and problems associated with the Lead PI?

Stakeholder: The goal is 'Assuring Lead PI'. The problem research don't have Lead PI'

Analyst: What are the goals and problems associated with the 'Recruits Non Lead PI'?

Stakeholder: The goal is 'Assuring the recruiting done by the lead PI'. The problem is 'Investigators aren't recruited

Analyst: What are the goals and problems associated with the 'Fills Document'

Stakeholder: The goal is 'Controlling That Non Lead PI document'.

Analyst: What are the goals and problems associated with the 'Reviews Document'

Stakeholder: The goal is 'Guaranteeing that Non Lead PI reviews Document'.

Analyst: What are the goals and problems associated with the 'Signs Document'

Stakeholder: The goal is 'Ensuring that Non Lead PI signs Document'.

Analyst: What are the goals and problems associated with the 'Submits Document'

Stakeholder: The goal is 'Controlling that Non Lead PI Submits Document'.

Analyst: What are the goals and problems associated with the 'Provides Study Information'

Stakeholder: The goal is 'Ensuring that Non Lead PI provides study information'.

Analyst: What are the goals and problems associated with the 'Notifies Review Authority'

Stakeholder: The goal is 'Guaranteeing that Non Lead PI notifies Review Authority'.

Analyst: What are the goals and problems associated with the 'Confirms Review.'

Analyst: What are the goals and problems associated with the 'Forwards Correspondence'

Stakeholder: The goal is 'Ensuring that Non Lead PI Forwards Correspondence'.

Analyst: Thank you for your valuable information. We will be in contact in order to clarify any doubts that may arise in this process.

Stakeholder: Thank you. I'll be in touch.

Dentro de todo el diálogo de analista e interesado solo se observan las siguientes preguntas con la estructura especificada para los eventos:

Pre1. Analyst: What does the Ceding IRB Administrator need in order to accomplish the 'Documents the Review Authority Status'?

Stakeholder: He just needs to have Review Authority Status= 'Ceded'

Pre2. Analyst: What does the Reviewing IRB Administrator need in order to accomplish the 'Notifies of Review Authority Status'?

Stakeholder: He just need to have Review Authority Status= 'Not Ceded'

Pre3. Analyst: What does the user need in order to accomplish the 'Modifies Protocol'?

Stakeholder: He just need to have Protocol Modification Status= 'Approved'

Pre4. Analyst: What does the user need in order to accomplish the 'Notifies Status'?

Stakeholder: He just need to have Status= 'Complete'

En estas preguntas y respuestas se cumplen la regla **CR 1.1<Quest>::=**<"What does the">+<person>+<"need in order to accomplish the">+<Concept>+<Verb1>+<"?">y la regla **CR 1.2.2 <ConditionalAnsw>::=**<"He just needs to have">+<Cond-Complete> lo que significa que estos 4 enunciados son eventos disparadores tipo condicional (véase la Tabla 12 para relacionar las reglas).

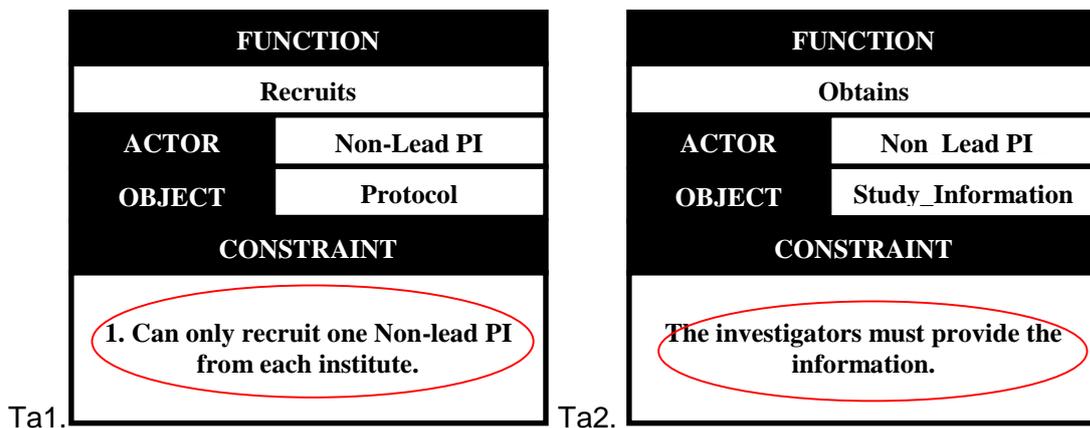
5.2 Tarjetas de educación de requisitos

Para que haya consistencia los eventos que se presenten en un artefacto se deben incluir en los demás; en este caso, se presentan varias inconsistencias (véase la Figura 23): las restricciones de las tarjetas (Ta1) y (Ta2) no se encontraron en el diálogo controlado.

La tarjeta (Ta3) representa las preguntas (Pre1) y (Pre2) y la tarjeta (Ta4) incluye la pregunta (Pre3) del diálogo controlado. Las tarjetas (Ta4) y (Ta5) no provienen de las preguntas del dialogo controlado relacionadas con eventos.

El verbo utilizado en la función es diferente del que se utiliza en el dialogo controlado.

HMO Research Network: Dominio del problema, Non-Lead_PI: investigador



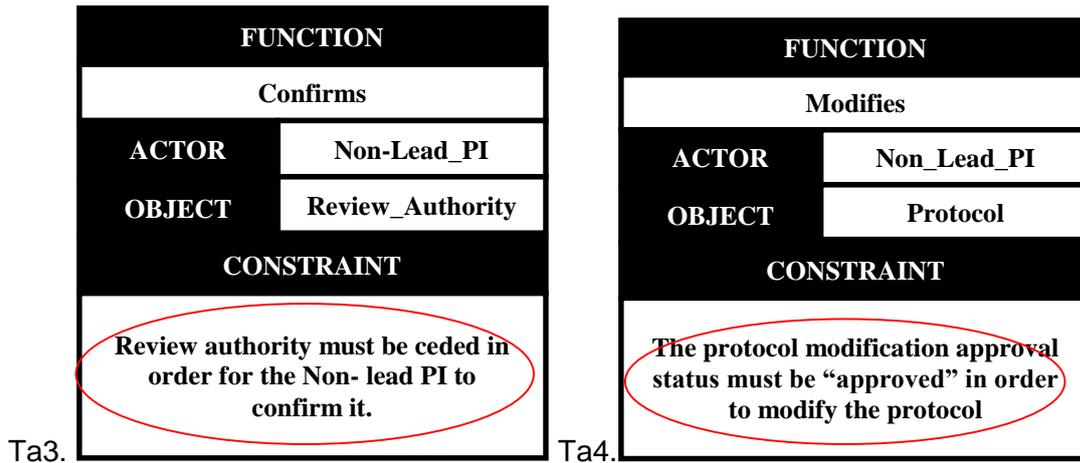


Figura 29. Tarjetas de educación de requisitos

Se sugiere para la representación de las tarjetas de educación de requisitos lo siguiente: aplicar la regla CR2 donde la restricción se debe representar como se va a mostrar en los demás artefactos. De esta manera, se toman a modo de ejemplo las tarjetas (Ta3) y la (Ta4). Se deben utilizar los verbos que incluyen a partir del dialogo controlado para que haya consistencia. En el caso de la tarjeta Ta3 no se incluía, sino que debería ser de la siguiente forma:

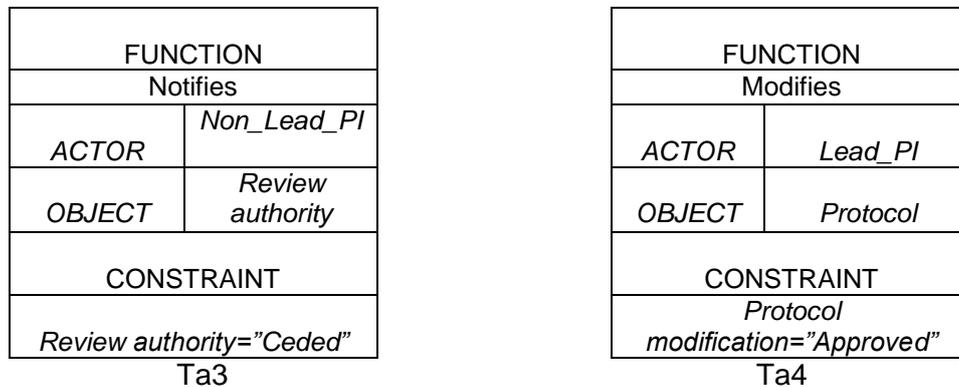
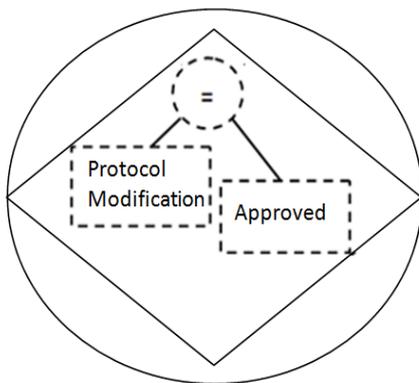


Figura 30. Corrección Tarjetas de educación de requisitos

5.3 Esquema preconceptual

Al observar el esquema preconceptual que realizaron los estudiantes (véase la Figura 26), tomando las tarjetas (Ta3) y (Ta4), se observa que se representaron debidamente los eventos disparadores tipo condicional. Sin embargo, aplicando la regla CR 3.1 se verifica que falta el círculo que denota a los eventos. Debería ser de la siguiente manera:

Evento condicional desde Ta3.



Evento condicional desde Ta4

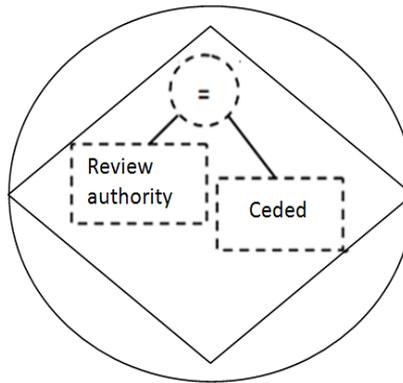


Figura 31. Corrección de Eventos en diagrama preconceptual

Se observa la correcta representación de los eventos disparadores al aplicar la regla CR3. También se utilizan los verbos apropiados para los eventos. No se presentan eventos de resultado.

77 Un mecanismo de consistencia en los eventos disparador y de resultado para los artefactos de UNC-Method

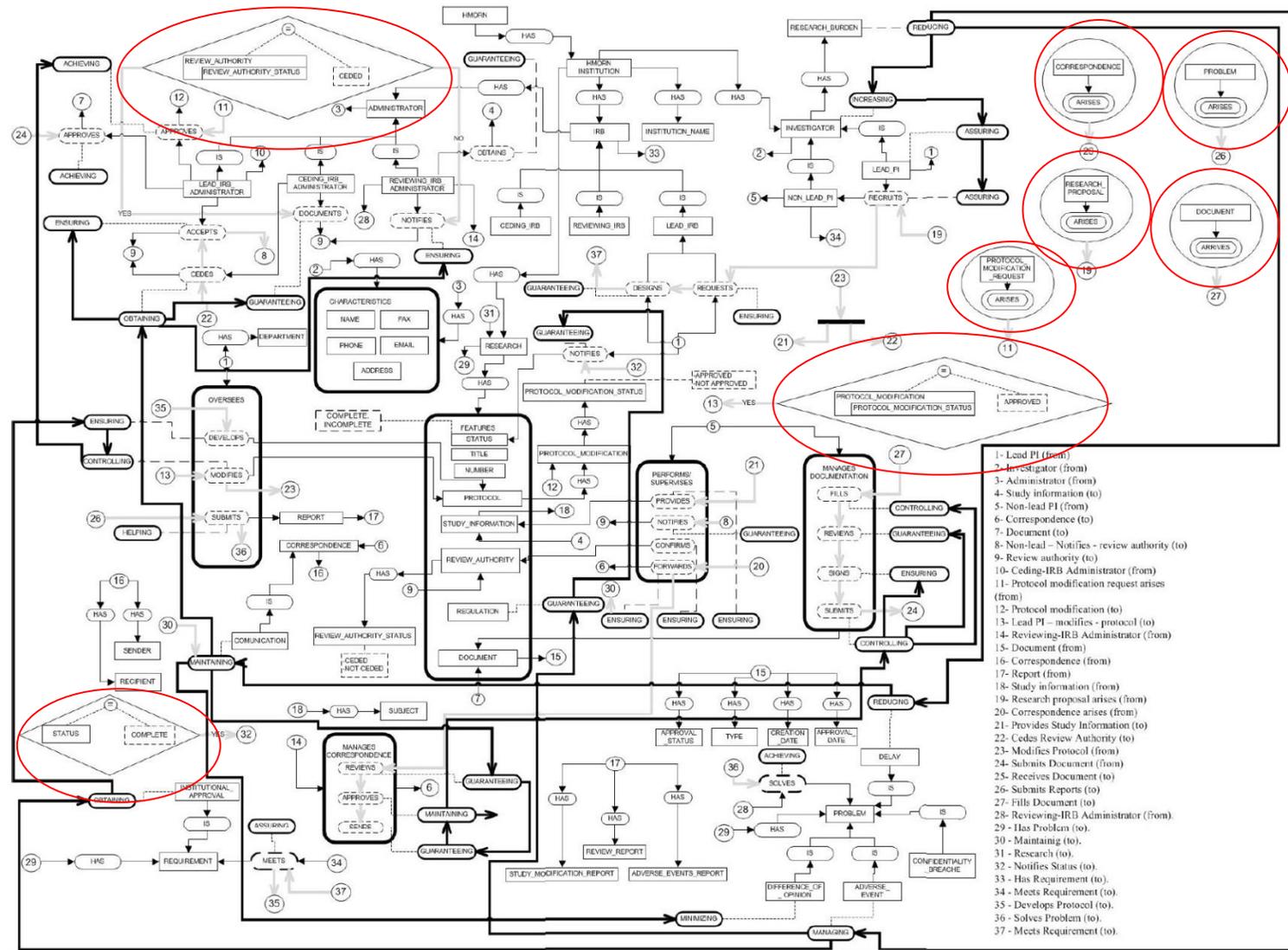


Figura 32. Esquema preconceptual

5.4 Diagrama de procesos

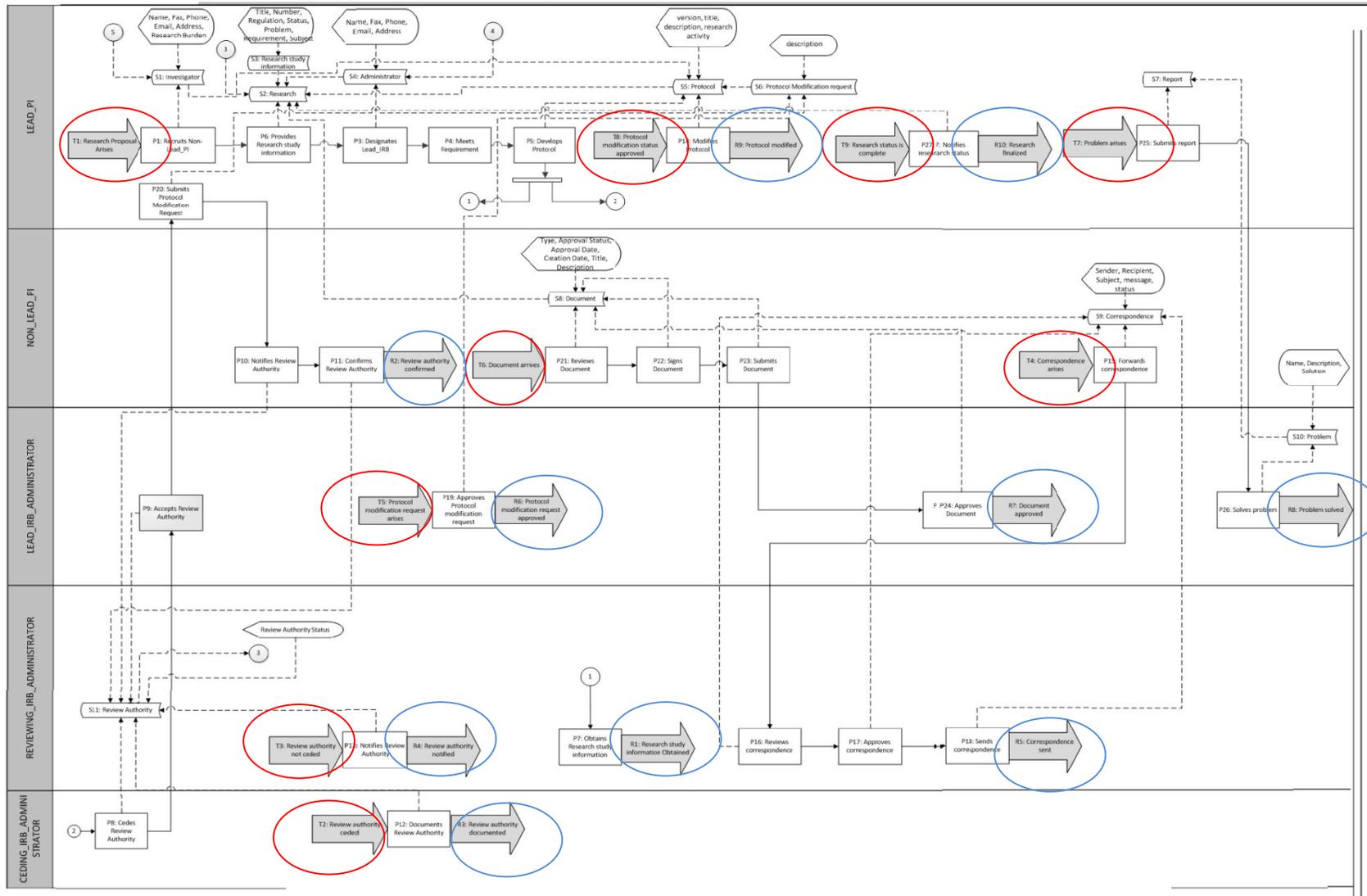


Figura 33. Diagrama de procesos

Desde el dialogo controlado se presentan inconsistencias en la cantidad de eventos ya que en este punto del diagrama de procesos, se están representando 9 eventos disparadores, en el diálogo controlado sólo se perciben 4 y en el esquema preconceptual se representan 8. Para lograr la consistencia de los artefactos, se deberían representar en cada artefacto en igual cantidad.

Los eventos disparadores y de resultado se representan correctamente de acuerdo con el cumplimiento de las reglas CR5 y CR7.

Recordando que este evento proviene del diálogo controlado y el esquema preconceptual como un condicional, se debe representar de la misma manera. Así, sobraría el proceso P11 de confirmación y el evento R2 *review authority notified* y T2 *review authority ceded* hacen referencia al mismo evento. Sólo se deberán colocar los que se presenten en los anteriores artefactos. De esta manera, no se trata de un evento de resultado sino de un disparador y la notación apropiada sería:

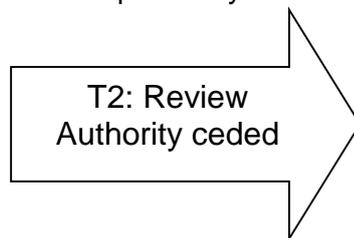


Figura 34. Corrección de eventos en diagrama de procesos

El evento R6 posee el mismo problema y se debería representar como evento disparador y no como resultado ya que así se definió en los anteriores artefactos.

5.5 Tabla explicativa de procesos

El objetivo de la tabla explicativa es relacionar los procesos con los objetivos problemas y eventos. Para estos últimos se verifica que su cantidad representada en el diagrama de eventos sea la misma en la tabla explicativa.

Tabla 13. Tabla explicativa de procesos

Name	Goal	Duration/ Frequency	How?/Where?	Problems	Business Rules	Events
P1: Recruits Non Lead_PI	G3: Increasing Investigators R2: Ensuring that the Non-lead PIs are recruited	One week/once time	The research needs some investigator that performs it on his institutions; the Lead PI recruits them and registers his personal information in his office.	C1: Few non lead PI are recruited.	BR003	T1: Research proposal arises
P2: Creates IRB Administrator	G7: Assuring Lead IRB. E1: Ensuring that the lead IRB is requested.	One day/One time	The system administrator adds the IRB Administrator user accounts and their information to the database using the application.	C6: The Lead IRB is not requested C5: Lead PI is not designated.		T16: User accounts creation request arrives.
P7: Obtains Research Study Information	G11: Guaranteeing Requirement R8: Keeping that the research study information is obtained.	5 minutes/ever y week	The reviewing IRB receives the information about the research in the meeting room.	C16: Meagre Research Study Information.	BR003	R1: Research Study Information obtained
P11: Confirms review authority	G2: Managing review authority E2: Ensuring that the review authority is confirmed.	5 minutes/once time	The Non lead need to confirm that the review authority was ceded to the Lead IRB in his institution in meeting room.	C10: Hardly any communication is effective. C18: The review authority is not ceded	BR001 BR002	R2: Review authority confirmed
P12: Documents review authority	G2: Managing review authority R1: Keeping that the review authority is documented.	1 hour/once time	In some moment in the research one institution have the possibility to cede the review authority to Lead IRB, when this happens the administrator has to document his decision he does it on his workplace.	C17: The review authority is not documented. C18: The review authority is not ceded	BR001	T2: Review authority ceded R3: Review authority documented
P13: Notifies review authority	G2: Managing review authority E3: Keeping that the review authority is notified.	5 minutes/once time	If the decided to not cede the review authority in all the research the administrator has to notifies his decision to all HMORN institutions in the meeting room.	C18: The review authority is not ceded C10: Hardly any communication is effective.	BR002	T3: Review authority not ceded R4: Review authority notified
P14: Modifies protocol	G13: Taking care of protocol. R15: Controlling that the protocol is modified.	One week/once time	When the Lead PI think that is necessary to modify the protocol and he has the approval of LEAD IRB administrator he can modify the research protocol in the workplace.	C7: Protocol has some errors. C8: Insufficient control in protocol modifications C9 The protocol isn't Correctly developed	BR003, BR005	T8: Protocol modification status approved R9: Protocol modified
P15:	G10: Managing	10	The research network	C15: Rarely the	BR003	T4:

Forwards correspondence	correspondence . R11: Improving that the correspondence is forwarded.	minutes/ever y week	need to be in constant communication, so the Non Lead PI need to forward the correspondence to the appropriate IRB administrator in his office.	correspondence is sent.		Correspondence arises
P18: Send correspondence	G10: Managing correspondence . E5: Improving the sending of the correspondence .	10 minutes/ever y week	When the reviewing IRB administrator knows how the research is going in other institutions, he sends a mail to other institutions in his office.	C15: Rarely the correspondence is sent.	BR003	R5: Correspondence sent
P19: Approves protocol modification	G13: Taking care of protocol R12: Controlling that the protocol modification is approved.	5 minutes/once time	In some occasions, the protocol needs to be modified, the Lead IRB needs to evaluate if the modification is necessary and he approve it in his workplace.	C8: Insufficient control in protocol modification. C9: The protocol isn't correctly developed.	BR003	T5: Protocol modification arises R6: Protocol modification request approved
P21: Reviews document	G14: Taking care of document. E7: Guaranteeing that the documents are reviewed.	One hour/ever y month	The Non Lead PI have to ensure that the information in the documents is correct before send it to a superior in his workplace.	C14: Documents aren't filled.	BR004	T6: Document arrives
P24: Approves document	G14: Taking care of document. R10: Guaranteeing that the document is approved	30 minutes/ever y month	The Lead IRB give his approbation to the information given in the document in his workplace.	C14: Documents aren't filled.	BR004	R7: Document approved
P25: Submits report	G4: Managing adverse events. R6: Improving that the report is submitted.	One hour/ever y week	When an adverse event occurs the Lead PI needs to do a report of them in the meeting room.	C13: The reports aren't submitted.		T7: Problem arises
P26: Solves problem	G4: Managing adverse events. R3: Helping that the problems are solved.	1 day/ever y week	The Lead IRB is the chosen one to solve problems in each institution, he receives the report and then he analyze and try to solve this problem in his office.	C13: The reports aren't submitted.		R8: Problem solved
P27: Notifies research status	G11: Guaranteeing Requirement R19: Assuring that the research status is notified	One time/ever y week	The Lead PI needs to inform to all HMORN how is going the investigation in the meeting room.	C12: There are delays in communication.		T9: Research status is complete R10: Research finalized
P28: Creates Lead PI	G3: Increasing investigators	One day/One time	The system administrator adds the lead PI user accounts and their information to the	C1: Few Non Lead PI are recruited		R16: User accounts created.

			database using the application.			
P29: Modifies IRB Administrator	G7: Assuring Lead IRB.	1 Hour/Every week	The system administrator modifies the data stored for the IRB administrator using the application.	C5: The Lead IRB is not designated		T10: IRB Administrator modification request arrives. R10: IRB Administrator modified.
P30: Deletes IRB Administrator	G7: Assuring Lead IRB.	30 minutes/Every week	The system administrator deletes the data stored for the IRB administrator from the database using the application.	C5: The Lead IRB is not designated		T11: IRB Administrator deletion request arrives. R11: IRB Administrator deleted.
P31: Modifies Lead PI	G3: Increasing investigators	1 Hour/Every week	The system administrator modifies the data stored for the Lead PI using the application.	C1: Few Non Lead PI are recruited		T12: Lead PI modification request arrives. R12: Lead PI modified.
P32: Deletes Lead PI	G3: Increasing investigators	30 minutes/Every week	The system administrator deletes the data stored for the Lead PI from the database using the application.	C1: Few Non Lead PI are recruited		T13: Lead PI deletion request arrives. R13: Lead PI deleted.
P33: Modifies Non-lead PI	G3: Increasing investigators	1 Hour/Every week	The system administrator modifies the data stored for the Non-lead PI using the application.	C1: Few Non Lead PI are recruited		T14: Non-lead PI modification request arrives. R14: Non-lead PI modified.
P34: Deletes Non-lead PI	G3: Increasing investigators	30 minutes/Every week	The system administrator deletes the data stored for the Non-lead PI from the database using the application.	C1: Few Non Lead PI are recruited		T15: Non-lead PI deletion request arrives. R15: Non-lead PI deleted.

Continúa la misma inconsistencia de los anteriores artefactos. En este caso, se denotan 15 eventos disparadores y 15 de resultado, cuando en el diagrama de procesos solo hay 19 eventos.

Exclusivamente, los mismos eventos que haya en el diagrama de procesos, se colocan en la tabla explicativa de procesos, ya que el objetivo de la tabla es explicar cada proceso definido en el diagrama de procesos.

5.6 Grafo de interacción de eventos, EIG

Este importante aclarar que este artefacto se incluye en UNC-Method desde el semestre II-2012, el cual se compartió mediante un juego en el aula de clase.

Los autores procuraron un intento de tener la consistencia de los artefactos, ya que tiene 19 eventos que corresponden a los del diagrama de procesos por lo tanto tiene los mismos errores que ese artefacto. Como se observa en la figura 35 se analiza una parte del grafo de interacción de eventos donde se señala, que el mismo evento se representa como evento disparador y como evento de resultado. La correcta representación se concibe como un evento disparador de tipo condicional ya que Los demás aspectos cumplen con las reglas CR8, CR9 y CR10.

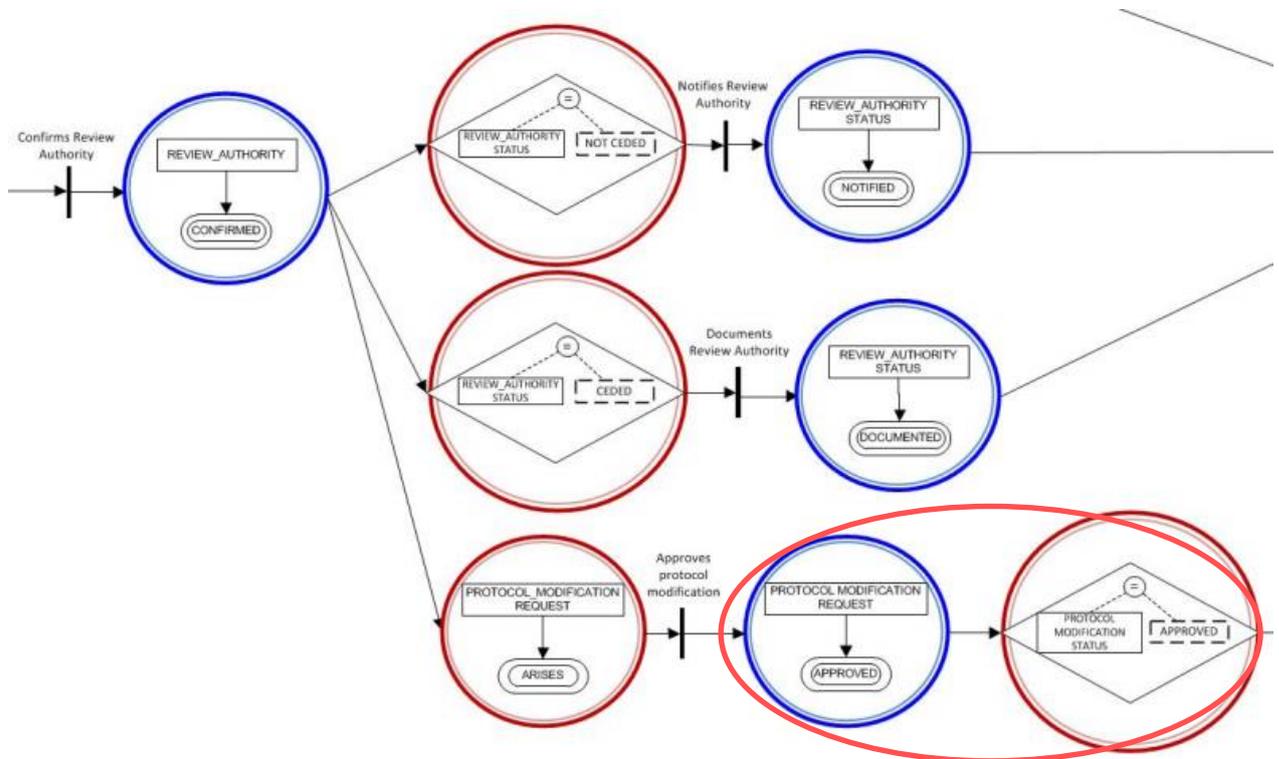


Figura 35. Parte del EIG

5.7 Diagrama de máquina de estados

Para este artefacto se pretende adoptar la notación que presenta Fowler en el diagrama de estados de UML con el objetivo de evidenciar los eventos disparadores de manera correcta. Actualmente se presenta de esta manera:

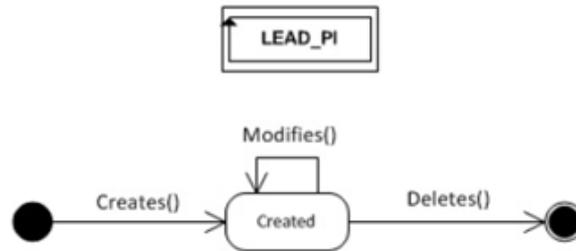


Figura 36. Diagrama de máquina de estados

Se propone utilizar la estructura de Fowler: Evento [condición] acción, como se presenta en la regla de consistencia CR11. Teniendo en cuenta que no siempre tienen que estar las tres partes presentes. El evento que se representa en el diagrama de máquina de estado es un evento disparador.

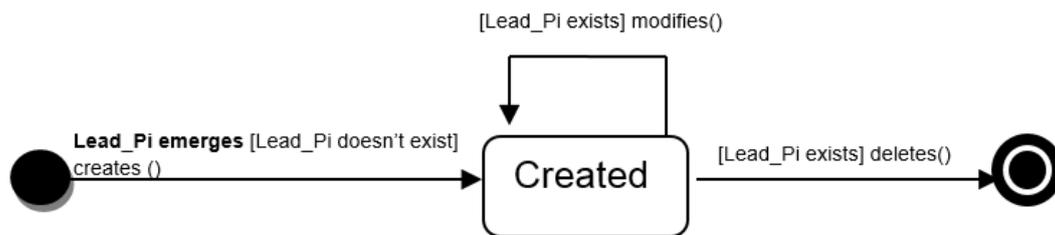


Figura 37. Corrección de eventos en diagrama de máquina de estados

Conclusiones

La consistencia es una de las características más importantes en ingeniería de software, pues su aplicación desde la definición de requisitos hasta la entrega del proyecto de software permite satisfacer las necesidades y expectativas del interesado. Los eventos disparadores y de resultado, cumplen un rol significativo dentro de la ingeniería de software, ya que generan un control en la ejecución de los procesos de negocio.

En esta Tesis de Maestría se caracterizaron los eventos disparadores y de resultado, en los artefactos de UNC-Method. A partir de esta caracterización, se propuso una definición de estructuras gramaticales (patrones lingüísticos de tipo léxico, sintáctico y semántico) que permiten diferenciarlos eventos de las acciones y operaciones que ejecutan los usuarios. Con estas estructuras gramaticales, se plantearon reglas de consistencia para cada uno de los artefactos que representan eventos, con el fin de contribuir al mejoramiento de su consistencia. Además, un artefacto para la representación de la interacción de eventos en UNC-Method, que permite observar la relación, secuencia y orden de ocurrencia de los eventos. De esta manera, las estructuras gramaticales, las reglas de consistencia y el artefacto permiten una completa representación de los eventos en las etapas del método.

Si bien varios de los elementos que se revisan y definen en esta Tesis se vienen trabajando de manera informal en el curso de ingeniería de requisitos de la

Universidad Nacional de Colombia, sede Medellín, se pudo evidenciar en el caso de estudio que la carencia de formalismo en la definición de las reglas de consistencia induce la aparición de errores en los diferentes artefactos. En el caso de estudio se reflejaron implícitamente algunos de los fundamentos del manejo de eventos que se formalizaron en esta Tesis. Sin embargo, se pueden apreciar múltiples errores en el trabajo de los estudiantes que se podrían omitir si se emplea de manera cuidadosa el mecanismo que se definió en esta Tesis. Así, este mecanismo puede constituir una herramienta útil para profesores, monitores y estudiantes de esta asignatura, pues de allí se puede extraer un listado de chequeo que posibilite la definición adecuada de los eventos en cualquiera de los artefactos de UNC-Method que los incluye. También, es una herramienta que se puede usar industrialmente para el proceso de educación de requisitos, pues podría minimizar el número de errores en que incurren analistas e interesados por el desconocimiento del significado preciso del término *evento* y su materialización para el desarrollo de aplicaciones de software.

Como posible trabajo futuro que se derive de esta Tesis de Maestría, se cuentan las siguientes líneas de investigación:

- Mejoramiento del prototipo para la validación de la consistencia de los eventos en UNC-Method: en este momento, el prototipo sólo maneja información textual y revela la información que debería aparecer en cada uno de los artefactos del método. En un futuro, este prototipo podría crecer hacia una herramienta CASE completamente orientada hacia el manejo de eventos a nivel gráfico inclusive.
- Análisis intensivo de las diferencias entre los diferentes verbos que representan eventos: si bien se estableció una caracterización inicial de los verbos que denotan eventos, el hecho de tener que incluir verbos con más de una valencia en los eventos se convierte en una fuente de investigación para precisar en qué momento tales verbos pueden constituir eventos y en qué momento no. La hipótesis inicial de un proyecto tal debería tomar en cuenta el alcance de la

aplicación de software que se está construyendo, pues, con base en los límites de la nueva aplicación, se podría establecer que una operación que escape al alcance del dominio que se trata de definir, puede constituir un evento para ese dominio.

- Exploración de otros artefactos que contemplen la representación de eventos disparadores y de resultado: se hizo una revisión intensiva de los artefactos de UNC-Method e, incluso, se propuso un nuevo artefacto, pero no se hizo énfasis en otros artefactos de diferentes metodologías que podrían ayudar en la representación de los eventos. Particularmente, el diagrama BPMN de RUP podría ser una opción interesante para trabajar, pues su riqueza sintáctica para el manejo de eventos sugiere nuevas oportunidades de exploración de situaciones eventuales.
- Formalización de los eventos en otras metodologías de orientación formal: particularmente, la metodología KAOS contiene diferentes artefactos formales que se complementarían muy bien con la definición de eventos.

Referencias

Belli.F., Nissanke N., Budnik, C.J. & Mathur, A. (2005). Test Generation Using Event Sequence Graphs. Technical Reports and Working Papers. Pederbon, Alemania.

Eloranta, L., Kallio, E. & Terho, I. (2006). A Notation Evaluation of BPMN and UML Activity Diagrams. Standard Document.
http://www.soberit.hut.fi/T-86/T-86.5161/2006/BPMN_vs_UML_final.pdf

Fowler, M. (2003).UML distilled: a brief guide to the standard object modeling language. Minneapolis, USA: New Softcover.

Galicia, S. (2000). Análisis sintáctico conducido por un diccionario de patrones de manejo sintáctico para lenguaje español. Tesis doctoral, Centro de Investigación en Computación, Instituto Politécnico Nacional, México DF, México.

Gelbuck, (2009). Análisis morfológico automático del española través de generación. México DF, México.

Gerring, J. (2007). "The mechanistic worldview: thinking inside the box". British Journal of Political Science 38: 161-179.Cambridge University Press. United Kingdom.

Guerra, E. & De Lara, J. (2007). Event-driven grammars: relating abstract and concrete levels of visual languages. Software and Systems Modeling. vol.6, no.3, pp.317-347.Heidelberg, Alemania.

Grgec, M. & Muzár, R. (2007). Role of UML sequence diagram constructs in object lifecycle concept. Journal of information and organizational sciences, vol. 31, no. 1.pp63-74. Varazdin, Croacia.

Hernández, U. & Álvarez, F. (2010).Use processes—modeling requirements based on elements of BPMN and UML Use Case Diagrams.2nd International Conference on Software Technology and Engineering (ICSTE), pp.36-40.San Juan, Puerto Rico.

Hoey, M. (1991) Patterns of Lexis in Text. United Kingdom: Oxford.

Katayama, T.; Furukawa, Z. & Ushijima, K. (1995), Event interactions graph for test-case generations of concurrent programs, Software Engineering Conference. Proceedings Asia Pacific, vol., no., pp.29-37. Fukuoka, Japón.

Jaramillo, A., Zapata, C. M. y Arango, F. (2005). Una propuesta para el reconocimiento semiautomático de operaciones utilizando un enfoque lingüístico. *Revista Facultad de Ingeniería Universidad de Antioquia*. Vol. no.34. pp. 42-51. Medellín, Colombia.

OMG, Object Management Group. (2009). *Business Process Model and Notation BPMN. Standard Document*. <http://www.omg.org/spec/BPMN/1.2>.

Oracle. (2005). Documento número: B14220-02. Documento estándar. http://docs.oracle.com/cd/B19306_01/server.102/b14220/triggers.htm.

Osorio, F. (2008). *Lógica y programación orientada a los objetos: un inicio al desarrollo de software*. Medellín, Colombia.

Owen, M. & Raj, J. (2003). *BPMN and Business Process Management*. Popkin Software. Technical report. http://www.omg.org/bpmn/Documents/6AD5D16960.BPMN_and_BPM.pdf

Lerner, B.S., Christov, S., Osterweil, L.J., Bendraou, R., Kannengiesser, U., y Wise, A. (2010) Exception Handling Patterns for Process Modeling, *IEEE Transactions on Software Engineering*, vol.36, no. 2, pp.162-183. South Hadley, USA.

Memon, A.M., Pollack M.E. y Soffa M.L. (2001) Hierarchical GUI Test Case Generation Using Automated Planning". *IEEE Trans. Software Engineering*, vol. 27, no. 2, pp. 144-155. Pittsburgh Univ., Pennsylvania, USA.

Memon, A.M. (2007). An event-flow model of GUI-based applications for testing. *Software testing, verification and reliability*, vol. 17, pp. 137–157. Maryland, USA.

Pan, Y., Zhou & Wang. L. (2009). Modeling Internet Applications with Event Graphs. *Sixth International Conference on Fuzzy Systems and Knowledge Discovery*. vol. 1, pp. 124-128. New York, USA.

Samuel, P., Mall, R. & Bothra, A. (2008). Automatic test case generation using unified modeling language (UML) state diagrams. *Software, IET*, vol.2, no.2, pp.79-93. 2008.

Singh, S., Gupta, R. & Sabharwal, S. (2009). Automatic extraction of events from Textual Requirements specification. *Nature & Biologically Inspired Computing. NaBIC 2009 World Congress*, pp.415-420. Coimbatore, India.

Swinnen, J., Vanhoof, K. & Hannes, E. (2010). Querying event logs: Discovering non-events in event logs. *2010 International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, vol., no., pp.349-354, 15-16. Hangzhou, China.

Urquía M. (2003). *Simulación. Texto Base de Teoría*". Departamento de Informática y Automática. Escuela Técnica Superior de Ingeniería Informática, UNED. p. 321. Madrid, España.

Vargas, F.A. (2010). Método para establecer la consistencia de los problemas en el diagrama causa efecto con el diagrama de objetivos de Kaos. Maestría thesis, Universidad Nacional de Colombia sede Medellín. <http://www.bdigital.unal.edu.co/2025/#sthash.2LJCTRLI.dpuf>

Vendler, Z. (1957). *Verbs and Times*. Linguistics in Philosophy. Nueva York, USA: Cornell University Press.

Watson & MaCabe. T.J. (1996). *Structured testing: a testing methodology using the cyclomatic complexity metric*. Technical Report NIST 500-225.

Weinbach, N. y García, A. (2004). *Una Extensión de la Programación en Lógica que incluye Eventos y Comunicación*. VI Workshop de Investigadores en Ciencias de la Computación. (WICC 2004). pp. 379-383. Neuquén, Argentina.

White, S.A. & Derek. M. (2009). *Guía de Referencia y Modelado BPMN*.

Yuan, X. & Memon. A. (2010 a). *Generating Event Sequence-Based Test Cases Using GUI Runtime State Feedback*. IEEE Trans. Software Engineering, vol.36, no.1, pp.81-95. Maryland, USA.

Yuan, X. & Memon. A. (2010b). *GUI interaction testing: Incorporating event context*. IEEE Trans. Software Engineering, vol.99. Maryland, USA.

Yuan, X. & Memon. A. (2010 c). *Iterative execution-feedback model-directed GUI testing*. Information and Software Technology, Vol 52. pp. 559–575. Maryland, USA.

Zapata. C.M. y Arango, F. (2006). *UN-MÉTODO para la Elicitación de Requisitos de Software*. Medellín, Colombia.

Zapata C.M y Carmona, M. (2010) *Un Modelo de dialogo para la Educción de requisitos de software*. Dyna. Medellín, Colombia.

Zapata C.M y Giraldo, G. (2009). *El juego del dialogo de edución de requisitos de software*. Avances en sistemas e Informática, Vol. 6, N°1, pp. 105-114. Medellín, Colombia.

Zapata C.M. (2010). *Computational Linguistics for helping Requirements Elicitation: a dream about Automated Software Development*, Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas, pp. 117–124, Los Angeles, USA.

Zapata, C.M. (2012) *The UNC-Method revisited: elements of the new approach*. Saarbrücken, Alemania: Langton.

Zowghi & Gervasi, V. (2002). *The Three Cs of Requirements: Consistency, Completeness and Correctness*. En: Proceedings of the 8th International Workshop on Requirements Engineering: Foundation for Software Quality, (REFSQ'02), Essen, 2002. pp. 155-164. Essen, Alemania.