

Hacia una taxonomía de la actividad (TXA) a partir de la separación de niveles de abstracción en casos de uso*

María L. Villegas**
César A. Collazos***
William J. Giraldo****
Juan M. González*****

Recibido: 07/12/2015 • Aceptado: 29/04/2016
DOI: 10.22395/rium.v16n30a5

Resumen

Los casos de uso (CU), tradicionalmente, han sido utilizados por equipos de desarrollo de software como una herramienta técnica y administrativa. Desde cada una de estas perspectivas, es necesario tener un conjunto de atributos a gestionar, ya que la información que se captura para el desarrollo de un producto software contiene información tanto del sistema, como de la organización. Si toda esta información pretende ser capturada a través de los CU, probablemente, lo que resulta es un aumento significativo de la complejidad de su especificación, lo cual genera cierto nivel de confusión a la hora de utilizarlos. En este trabajo nos proponemos la separación de elementos de negocio y técnicos, al momento de elaborar CU, usando la TxA. Presentamos la TxA en el modelado con CU para reducir su complejidad y principalmente para incrementar su efectividad y utilidad en lo que tiene que ver con la interacción humano computador.

Palabras clave: taxonomía de la actividad (TxA), interacción humano computador (HCI), casos de uso, desarrollo de sistemas interactivos

* El artículo se origina en una investigación de tesis doctoral titulada: Marco conceptual, metodológico y tecnológico para el modelado y ejecución de la actividad que soporta el diseño de la interfaz de usuario de sistemas interactivos, dentro del contexto de TD-MBUID y a partir de la infraestructura de CIAF. Fue financiada por Colciencias (Programa Nacional de Formación de Investigadores), Universidad del Quindío, Universidad del Cauca y AUIP (Asociación Universitaria Iberoamericana de Posgrado), Programa de Movilidad Académica Internacional. Se ejecutó entre julio de 2013 y diciembre de 2016. Participaron Universidad del Cauca, Colombia, Universidad del Quindío, Colombia, Benemérita Universidad Autónoma de Puebla, México.

** Msc, Phd (c), docente-investigador Universidad del Quindío, Grupo SINFOCI. Estudiante de Doctorado Universidad del Cauca, Grupo de investigación IDIS. Carrera 15 Calle 12 Norte, Armenia. Universidad del Quindío, Colombia. (+57) 6 7359300 ext 995. Dirección electrónica: mlvillegas@uniquindio.edu.co.

*** Phd, docente-investigador Universidad del Cauca, Director Grupo de Investigación IDIS. FIET Sector Tulcán, Popayán. Universidad del Cauca, Colombia. (+52)28209800 ext 2133. Dirección electrónica: ccollazo@unicauca.edu.co

**** Phd, docente-investigador Universidad del Quindío, Director Grupo de Investigación SINFOCI. Carrera 15 Calle 12 Norte, Armenia. Universidad del Quindío, Colombia. (+57)67359300 ext 995. Dirección electrónica: wjgiraldo@uniquindio.edu.co.

***** Phd, profesor titular A- Facultad de Ciencias de la Computación. Benemérita Universidad Autónoma de Puebla. Av. San Claudio y 14 Sur Edificio 104a - 208. Ciudad Universitaria. Puebla, México. (+52)229.55.00 doble extensión: 7241 + 178. Dirección electrónica: juan.gonzalez@cs.buap.mx

Towards an activity taxonomy (ATx) from separation of concerns in use case modeling

Abstract

Use cases (UC) have traditionally been applied by software development teams as a technical and administrative tool. From each of these perspectives, it is necessary to have a set of attributes to manage, because the information taken for the development of a software product contains information from both the system and the organization. If all this information is intended to be captured through the uc, probably what results is a significant increase in the complexity of its specification, which creates some level of confusion when using them. In this paper we propose the separation of business and technical elements, when drafting uc, using the ATx. We are introducing the ATx in modeling with uc to reduce its complexity and mainly to increase their effectiveness and usefulness in what has to do with the human computer interaction.

Key words: activity taxonomy (ATx), human computer interaction (HCI), use cases, interactive systems development.

INTRODUCCIÓN

La ingeniería de software (IS) se ha centrado, principalmente, en generar productos completamente funcionales, es decir, se ha enfocado en el aspecto de la funcionalidad, dejando un poco de lado otros atributos inherentes al desarrollo de sistemas interactivos, como la colaboración, la usabilidad, entre otros. La respuesta de la comunidad investigativa ante este enfoque netamente funcional ha sido la de los casos de uso (CU) e historias de usuario, los cuales permiten guiar el diseño y desarrollo de software, además de que se utilizan como punto de entrada a dichos procesos de desarrollo [1-5]. Partimos de los CU, porque se han definido como una excelente y poderosa forma de definir los requerimientos de comportamiento del software [6], porque permiten definir un proyecto y establecer un lenguaje común entre stakeholders. De esta forma, los CU han tomado un interés tanto administrativo, como técnico.

Desde el punto de vista de la HCI, el interés se centra más en capturar el modelo mental de los usuarios y en otros aspectos relacionados con la psicología cognitiva y al factor humano. Si toda esta información, tanto desde la Ingeniería de Software, como desde la HCI, pretende ser capturada a través de los CU, probablemente, lo que resulta es un aumento significativo de la complejidad de su especificación, lo cual genera cierto nivel de confusión a la hora de utilizarlos.

Concretamente, el CU es un elemento de modelado que ha sido ampliamente utilizado tanto en la ingeniería de software como en el diseño de interfaces de usuario [7]. Para comprender su estructura, es necesario dibujarlo en un diagrama. El CU se especifica mediante una especificación de CU, donde se indica la interacción y la labor del sistema [8]. Cada elemento de la especificación de un CU corresponde a un clasificador de información o token que describe la funcionalidad, y cada aproximación metodológica sigue o propone una estructura diferente para el artefacto “*especificación del CU*”. Ejemplos de clasificadores serían: “*Nombre del CU*”, “*flujo básico*”, “*precondición*”, “*postcondición*”.

Las discusiones en la literatura y la experiencia de los autores indican que, al parecer, existe un problema relacionado con la especificación de los CU, más que todo en lo que tiene que ver con la separación de niveles de abstracción. Resulta confuso modelar con CU cuando se mezclan *workers*, *entidades* y *actores de negocio*, *realizaciones*, *reglas*, *acciones*, componentes que hacen parte de los CU de negocio, con especificaciones relacionadas con la funcionalidad a nivel del sistema interactivo. Todo esto conlleva un aumento de la complejidad o mal uso tanto con el nivel del modelado de sus diagramas como en el nivel de su especificación, “*abuse cases*”, como lo denomina [6].

En este trabajo se propone estudiar los clasificadores o tokens que componen la especificación de los CU y cómo se organizan en torno a una *Taxonomía de la Actividad (TxA)*. La TxA estudia los clasificadores que hacen parte de las especificaciones de CU desde distintos puntos de vista, no solo desde el aspecto de la funcionalidad, sino también desde la usabilidad y del diseño de la interacción, para proveer un marco de organización de los clasificadores y analizar cómo a partir de ellos es posible crear artefactos, plantillas en el ámbito del desarrollo de sistemas interactivos. De esta forma podríamos determinar, por ejemplo, si la funcionalidad al nivel de negocio, debe ser especificada en una plantilla distinta a la funcionalidad al nivel de sistema.

Esta propuesta no pretende que las empresas desarrolladoras de software incorporen de lleno el diseño centrado en el usuario y en sus procesos, ni cambiar del todo los CU. Lo que se pretende es proporcionar un conjunto de técnicas y artefactos que incorporen aspectos de HCI en el desarrollo de software.

La sección 1 hace una breve descripción de los trabajos relacionados. La sección 2 describe los fundamentos conceptuales del trabajo. La sección 3 describe la TxA propuesta. La sección 4 presenta un caso de estudio. La sección 5 muestra los resultados y discusión. Por último, la sección 6 presenta las conclusiones.

1. TRABAJOS RELACIONADOS

La revisión de la literatura relacionada con taxonomías en el ámbito de la IS y de la HCI permitió realizar una recopilación de la información que clasifican, para compararla con la TxA propuesta. La descripción de las taxonomías estudiadas se encuentra detallada en [9]. El análisis del conjunto de propuestas permitió concluir que cada taxonomía, a diferencia de la TxA, es definida para un dominio muy específico. La mayoría de taxonomías se utiliza para comparar e identificar ventajas y carencias relacionadas con las características de propuestas que encajan en el conjunto de clasificadores definidos para dichas taxonomías. Adicionalmente, la mayoría de las propuestas analizadas no tienen definidos sus clasificadores de acuerdo con la separación en niveles de abstracción, como sí lo hace la TxA.

Concretamente, en el ámbito de la IS, y para el contexto de este trabajo, se observa, por ejemplo, cómo Jacobson [1] presenta una evolución de la técnica de CU enfocada en mejorar la gestión de proyectos de desarrollo de software, dejando un poco de lado los aspectos técnicos relacionados con la especificación de la funcionalidad de un producto software. Por otro lado, en el desarrollo centrado en el usuario se ha discutido mucho la eficacia y la utilidad de la técnica de especificación de CU [2, 3,10], a pesar de que resulta ser una práctica muy útil para capturar requerimientos funcionales, dirigir la integración con la interfaz de usuario, el diseño de la funcionalidad y la colaboración [4].

La aproximación propuesta en use-case 2.0 [1] aborda el problema de la complejidad en el modelado con CU incorporando el concepto “*use-case slice*” donde un “*slice*” es una fracción de un CU que se diseña, implementa y prueba de forma separada, pero desde una perspectiva administrativa. A diferencia de Jacobson [1], nuestra propuesta se centra en disminuir la complejidad en el modelado con CU desde una perspectiva técnica, aplicando una taxonomía de la actividad (TxA).

En la literatura de referencia se encuentran muchos trabajos que discuten la actividad desde la perspectiva de los CU, enfrentando los problemas de la complejidad que involucra su utilización. Por nombrar unos pocos, Nunes [2] asegura que los CU son deficientes para soportar actividades centradas en el usuario, son imprecisos en su definición, son inútiles, no proveen una forma eficiente para soportar los aspectos de usabilidad del sistema interactivo y no son una forma generalizada de modelado de tareas. Lozano [3] critica el lenguaje UML porque no aborda el modelado de la interfaz de usuario, a los casos de uso porque no permiten capturar los requisitos no funcionales y menciona que no permiten capturar las necesidades del usuario final de una aplicación informática. A partir de estas afirmaciones surgen preguntas como: los casos de uso son ¿buenos, malos, útiles?, ¿qué información capturan?, ¿qué información les falta por capturar?, etc. Discusiones y preguntas como éstas justifican un análisis del modelado de la actividad.

Se observa que los trabajos donde se realizan este tipo de discusiones no encuentran una base en la clasificación del modelado de la actividad. Particularmente, para los CU, el problema radica en cómo se utilizan, no en cómo están especificados. Como lo expone [8], los CU capturan realmente los requerimientos cuando se describen de una forma apropiada. No proveen detalles sobre interfaces externas, formatos de datos, reglas de negocio, etc. Constituyen solo una fracción de todos los requerimientos que deben ser capturados. Es decir, es imposible cubrir todos los aspectos en una sola especificación. Lo que se propone en este trabajo es aplicar la clasificación de la actividad en niveles de abstracción negocio y sistema para disminuir la complejidad de los CU, haciendo una separación en CU de negocio y CU de sistema, de acuerdo con [5, 6, 8, 11].

2. SEPARACIÓN DE NIVELES DE ABSTRACCIÓN COMO FUNDAMENTO CONCEPTUAL DE LA PROPUESTA

La abstracción es un medio cognitivo por el que los ingenieros, matemáticos y otros se ocupan de la complejidad, y ha sido reconocida como una habilidad crucial para los profesionales de la ingeniería del software [12]. Si bien lo que nos interesa en gran medida es el modelamiento de los sistemas, existen limitantes en la representación de la información que se modela en cuanto a niveles de abstracción, lenguaje y restricció-

nes en dichos niveles de abstracción. Por ejemplo, si tomamos la estructura básica de los niveles de transformación según el Framework de Zachman [13] para representar una arquitectura empresarial, conceptual, lógica y tecnológica, es posible diferenciar que cada nivel tiene sus propias restricciones. Por ejemplo, si nos preguntamos ¿Cuánto pesa una foto?, diferenciamos las restricciones que existen para cada nivel de abstracción:

- *Nivel conceptual*: la foto pesa en gramos (g), pesa el papel. ¿Cuántas fotos puedo almacenar?, (restricciones físicas).
- *Nivel lógico*: se define un arreglo con n elementos, fotos[1,...,n], (no hay restricciones lógicas).
- *Nivel tecnológico*: la foto pesa en kb, Mb,..., (restricciones de espacio en memoria).

La funcionalidad que se origina al nivel de negocio se va heredando hasta el nivel de la tecnología, así que en el nivel tecnológico es posible que falle un sistema si no se captura toda la información (actividad, forma, datos) a nivel de negocio. Adicionalmente, como lo recomienda Zachman [13], no se deben hacer especificaciones que aborden más de un nivel de abstracción.

Particularmente, en el desarrollo de sistemas software, el tipo de abstracción más comúnmente usada es la abstracción funcional y esto no ha cambiado mucho desde los años 70 [14].

3. PROPUESTA

Según lo mencionado en la sección 2, la abstracción involucra procesos cognitivos de clasificación y eliminación. Particularmente, el proceso de clasificación requiere de estructuras taxonómicas y de clasificadores que lo soporten. Este es uno de los principios en los que se fundamenta la TxA.

La TxA para sistemas interactivos se enfoca en identificar las categorías en las que se estructura toda la labor que se encuentra presente mientras se busca una meta de negocio [15]. Está fundamentada en los principios de la ingeniería dirigida por modelos (MDE) y en la clasificación del modelado de la actividad (CMA), que se refiere a encontrar las distintas clasificaciones de la actividad según los distintos aspectos del desarrollo de software, funcionalidad, colaboración, seguridad, desarrollo de la interfaz de usuario, etc. [4]. El modelado de la actividad es la práctica que se centra en la comprensión y representación de la naturaleza del trabajo con el fin de establecer una base para el diseño de un sistema que lo soporte [16].

En este trabajo se pretende mostrar cómo la TxA complementa la especificación de los CU, principalmente para identificar especificaciones que aportan a mejorar la HCI.

La taxonomía define un conjunto de clasificadores que representan cada aspecto del modelado de la actividad, llamado “*estructura taxonómica de la actividad*”. Define también un proceso de clasificación que explica cómo se clasifican los objetos bajo estudio dentro de la taxonomía. Por disponibilidad de espacio no se muestra la estructura taxonómica de la actividad en su forma genérica, sino que se ilustra de una vez con los resultados del caso de estudio en la sección 5.

3.1 Estructura taxonómica de la actividad

La estructura taxonómica de la actividad propuesta se divide en dos niveles de abstracción: negocio y sistema. El nivel de negocio se enfoca en capturar la esencia del sistema. Es un sistema libre de tecnología, en el que interesan las intenciones de los usuarios y el propósito fundamental del negocio. El nivel de sistema informático se centra en la definición de las actividades que están relacionadas con la interacción con el sistema [15].

Cada nivel de abstracción se clasifica en dos niveles de jerarquía entre las actividades: actividades de bajo nivel de granularidad y actividades del alto nivel de granularidad.

La clasificación por tipo de actividad intenta diferenciar la información que es esencial para el diseño de la interfaz de usuario y separarla de cualquiera otra información considerada útil para otros fines. Las actividades se clasifican según su tipo en: 1) actividades de interacción con los usuarios: actividades base y de soporte, y 2) actividades sin interacción con el usuario: actividades de gestión.

La participación entre los roles y las actividades es la siguiente: el actor de negocio y el actor de sistema intervienen en las actividades de tipo base y de soporte; el trabajador de negocio (*worker*) y el control del sistema intervienen en las actividades de tipo gestión. Al nivel de negocio, ambos, el actor de negocio y el worker intervienen en las actividades clasificadas como de inter-acción [17] y al nivel de sistema, el actor de sistema y el sistema, intervienen en las actividades clasificadas como interactivas.

3.2 Proceso de clasificación

El proceso consta de siete (7) pasos con los cuales se asegura que la TxA se aplica sobre casos de estudio con las características apropiadas y que responden a una serie de preguntas para clasificar la información capturada en su especificación. Los pasos que conforman el proceso son:

1. Relevancia
2. Caracterización básica
3. Niveles de abstracción
4. Tipos de actividad
5. Clasificación a nivel de negocio
6. Clasificación a nivel de sistema
7. Conclusiones

El detalle del proceso se encuentra descrito en [15]. Es importante resaltar que la información que se clasifica con la TxA solamente es la que se requiere que se transfiera al nivel de la tecnología.

4. CASO DE ESTUDIO

En esta sección se toma el ejemplo de la especificación del curso básico de acción para el caso de uso: “*inscribirse en una universidad*”, tomado de Ambler [18] (tabla 1). Se puede observar que desde el paso 1 hasta el paso 3 la especificación se describe al nivel de “*negocio*”. A partir del paso 4 se mezclan los niveles de abstracción “*negocio*” y “*sistema*”. Se contradice lo expuesto en la sección 2.

Tabla 1. Ejemplo de especificación del CU

<i>Caso de uso: inscribirse en una universidad</i>
<p>Curso Básico:</p> <ol style="list-style-type: none"> 1. Un aspirante desea inscribirse en la universidad. 2. El aspirante toma el formato UI13 Formato de Aplicación para la Universidad y lo diligencia. 3. El registrador inspecciona el formato visualmente y determina que el formato ha sido diligenciado apropiadamente. [Curso Alternativo B: Formatos Diligenciados Inapropiadamente]. 4. El registrador da click sobre el icono Crear Estudiante. 5. El sistema muestra la Pantalla UI89 Crear Estudiante y el registrador ingresa el nombre, dirección y número de teléfono del aspirante. [Punto de Extensión: UC34 Realizar Control de Seguridad]. 6. El sistema determina que el aspirante no existe dentro del sistema de acuerdo a BR37 Criterios de Coincidencia Potencial para Estudiantes Nuevos. [Curso Alternativo F: Estudiantes Parecen Existir Dentro del Sistema]. 7. El sistema determina que el aspirante está en la lista de aspirantes elegibles. [Curso Alternativo G: Persona no es Elegible para Inscribirse].

Caso de uso: inscribirse en una universidad

8. El sistema agrega al aspirante a sus registros. El aspirante es ahora considerado como estudiante.
9. El sistema calcula el pago inicial requerido de acuerdo a BR16 Calcular Cuotas de Inscripción y muestra la Pantalla UI15 Resumen de Matrícula.
10. El registrador pide al estudiante pagar el pago inicial de acuerdo a BR 19 Opciones de Pago de Matrícula.
11. El estudiante paga la cuota inicial. [Curso Alterno D: El Estudiante no Puede Pagar en Este Momento].
12. El sistema imprime un recibo.
13. El registrador da el recibo al estudiante.

Fuente: [18]

Existen muchos ejemplos similares a este, y como lo indica Ng [5], con frecuencia el análisis salta directamente a las especificaciones del sistema sin poner las necesidades dentro de algún contexto de negocio. Esto resulta en una tendencia a “olvidar” las responsabilidades que dichos sistemas tienen en el negocio.

Desde el punto de vista de la HCI, una actividad clave es la definición de las tareas de negocio y su interpretación a través del escenario sobre el cual va ser implementado. De este modo el usuario puede diferenciar información sobre, por ejemplo, feedback y awareness de negocio y de sistema.

4.1 Aplicación de la solución propuesta al caso de estudio

El principal interés de este trabajo es aplicar una clasificación de la actividad al menos en niveles de abstracción en el modelado con CU, para disminuir el nivel de complejidad en el modelado con esta técnica y evitar el procesamiento de la actividad como una unidad monolítica.

A continuación se muestra cómo se construye el modelo de CU separando niveles de abstracción para el ejemplo del CU “*inscribirse en una universidad*”.

4.1.1 Modelado de negocio

En el ejemplo del CU mencionado, “*inscribirse*” se toma como una tarea de negocio. El humano siempre la va a hacer, aunque exista tecnología o no. Los CU de negocio (figura 1a, tabla 2) permiten el análisis de requerimientos del negocio, mientras que las realizaciones (figuras 1b y 2) y los diagramas de actividad (figura 3) corresponden al diseño del negocio.

Un trabajador de negocio (*worker*) es una abstracción que permite orientar la trazabilidad desde el nivel de negocio hasta el nivel de sistema [4]. La relación entre

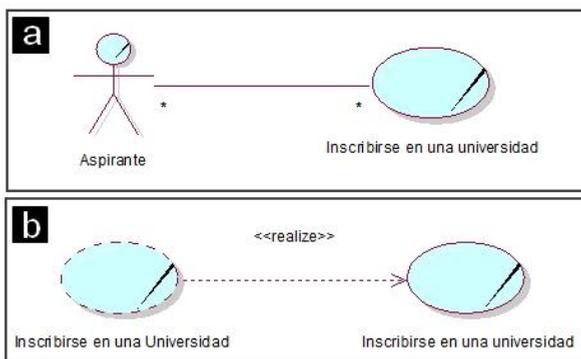


Figura 1. Diagrama de CU de negocio del ejemplo.
Fuente: elaboración propia

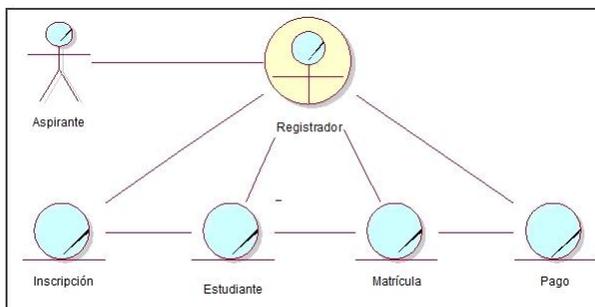


Figura 2. Realización de CU de negocio del ejemplo.
Fuente: elaboración propia

Tabla 2. Especificación del CU de negocio del ejemplo

<i>CU de negocio: inscribirse en una universidad</i>
1. Un aspirante desea inscribirse en la universidad.
2. El aspirante toma una copia del formato UI13 y la diligencia.
3. El registrador inspecciona el formato visualmente y determina que el formato ha sido diligenciado apropiadamente. [Curso Alterno B].
4. El registrador chequea si el aspirante ya existe en el Archivo de Estudiantes, de acuerdo a BR37, y determina que el aspirante no existe en el Archivo de Estudiantes. [Curso Alterno F].
5. El registrador chequea si el aspirante es elegible para inscribirse, de acuerdo a BRXX1 y determina que el aspirante es elegible para inscribirse. [Curso Alterno G].
6. El registrador toma el Formato XX1 y lo diligencia para registrar al aspirante.
7. El registrador guarda el Formato XX1 diligenciado en el Archivo de Estudiantes. El aspirante es ahora considerado como un estudiante.
8. El registrador le comunica al estudiante que ha quedado registrado.

8. El registrador le comunica al estudiante que ha quedado registrado.
9. El registrador toma el Formato XX2 y lo diligencia para calcular el pago inicial.
10. El registrador calcula el pago inicial.
11. El registrador pide al estudiante pagar el pago inicial.
12. El estudiante paga la cuota inicial. [Curso Alterno D].
13. El registrador diligencia un recibo de pago con la información del estudiante.
14. El registrador da el recibo al estudiante y deja una copia del recibo para el Archivo de Matriculas.

Fuente: elaboración propia

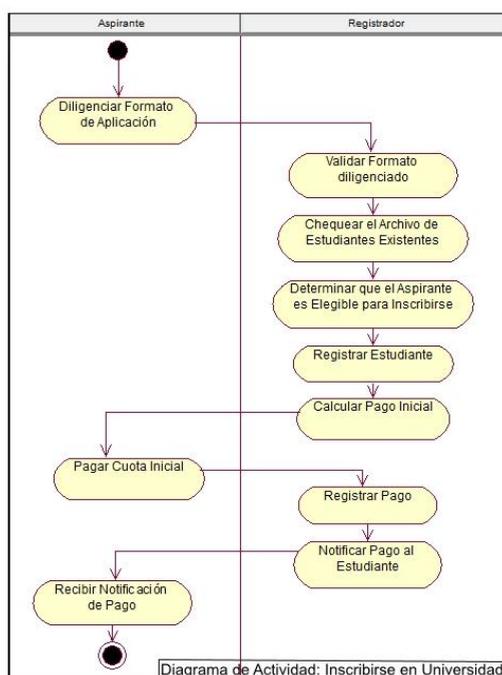


Figura 3. Diagrama de actividad: inscribirse en una universidad.

Fuente: elaboración propia

el *aspirante* y el *registrador* es necesaria para documentar los mecanismos de interacción que existen entre ellos (figura 2), elementos indispensables en la especificación de un sistema interactivo, desde la HCI. El diálogo que se lleva a cabo se divide en dos niveles: (1) *negocio*: entre cliente y *worker* y (2) *sistema*: entre actor del sistema y sistema informático (figura 4).

Realizando una comparación entre las especificaciones de los CU (tablas 1 y 2), se observa cómo en la primera especificación se mezclan los niveles de abstracción,

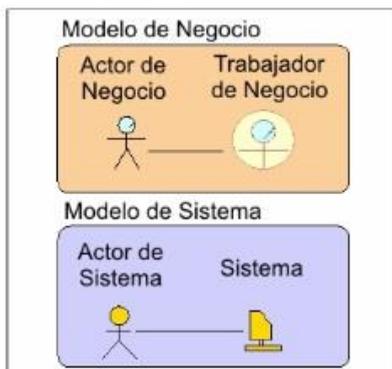


Figura 4. Realización del worker: registrador.

Fuente: elaboración propia

Especificación Tabla 2	Especificación Tabla 3
<p>Modelo de Negocio</p> <p>Pasos 1 a 3</p> <p>Modelo de Sistema</p> <p>Pasos 4 a 9</p> <p>Modelo de Negocio</p> <p>Pasos 10 y 11</p> <p>Modelo de Sistema</p> <p>Paso 12</p> <p>Modelo de Negocio</p> <p>Paso 13</p>	<p>Modelo de Negocio</p> <p>Pasos 1 a 14</p>

Figura 5. Comparación de especificaciones para el CU del ejemplo.

Fuente: elaboración propia

mientras que en la segunda, solo se especifica el nivel de negocio, aplicando la separación de niveles de abstracción recomendada (figura 5).

En este nivel el desarrollador se centra en el diseño del negocio de manera abstracta e independiente de la tecnología, lo cual permite reducir en cierto grado la complejidad en el modelado porque no se estarían mezclando especificaciones en los diferentes niveles de abstracción, tal y como se mencionó al inicio de esta sección. Las actividades de negocio representan las intenciones del cliente, las del trabajador del negocio y sus inter-acciones [17], manteniendo un alto nivel de independencia de la tecnología que se utiliza para ejecutar dichas actividades.

Las actividades de negocio se detallan a través de la especificación de una serie de pasos o acciones. Estos pasos son los de menor nivel de granularidad que es alcanzado en el modelado de la actividad a nivel del modelo de negocio. En un principio, en la especificación, no se sabe si un paso se llevará a cabo con apoyo de la tecnología o

será una actividad manual del usuario. La especificación en este nivel de granularidad es el primer paso en la identificación de los CU de Sistema.

4.1.2 Definición de CU de sistema

Se identifican las intenciones de los usuarios potenciales del futuro sistema. Estas intenciones se definen a través de una distinción entre las actividades que son llevadas a cabo por el usuario durante la realización de un proceso de negocio y las acciones que puedan ser automatizadas. Para la actividad “*registrar estudiante*”, las acciones de negocio se han marcado con <U>, <I> y <S> para indicar las que son ejecutadas por el usuario, las interactivas y las ejecutadas por el sistema, respectivamente (figura 6). Se presenta un diagrama de CU de Sistema que corresponde a la realización del *worker* “registrador”, pero que proviene exclusivamente de la actividad de negocio “*registrar estudiante*” que forma parte del proceso de negocio “*inscribirse en una universidad*” (figura 6). Se identifican entonces fácilmente los CU de sistema, aplicando la trazabilidad entre especificaciones en los dos niveles de abstracción.

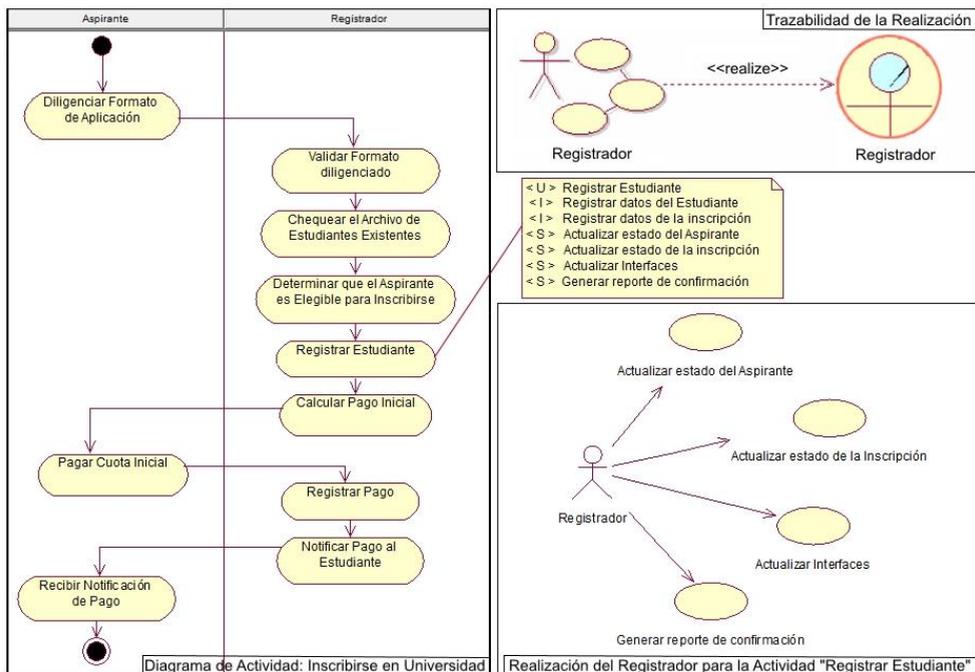


Figura 6. Relación entre actividades de negocio, pasos (acciones de negocio) y CU de sistema.

Fuente: elaboración propia

Por otro lado, el diagrama resultante para este ejemplo sugiere que el número de CU puede llegar a ser alto, si se cuenta con varios procesos de negocio y varias

actividades de negocio [4]. Probablemente entonces un desarrollador desearía especificar un solo caso de uso con las cuatro tareas del sistema que componen esta actividad para reducir el número total de CU. Sin embargo, estaría restringiendo la secuencia de actividades internas al modo en el que él entiende la actividad de negocio (no a como la entienden los usuarios) y estaría empaquetando comportamiento que posteriormente podría reutilizar.

A partir de este punto se continuaría el desarrollo del sistema a partir de los CU de sistema. Esta parte no corresponde con el alcance de este artículo.

5. RESULTADOS Y DISCUSIÓN

Retomando el ejemplo del CU “inscribirse en una universidad (tabla 1)”, y clasificando la información de la especificación, en la estructura taxonómica propuesta (tabla 3), se concluye que existen errores en la especificación solo por el hecho de mezclar niveles de abstracción. Se observa también que el CU captura menos información de la necesaria para modelar *tareas, roles, sus responsables y relaciones, feedback, awareness*. Desafortunadamente, en UML no se cuenta con elementos de modelado que permitan clasificar adecuadamente el modelado de la actividad para diferenciar cada uno de estos elementos. Particularmente, el modelo de negocio en UML utiliza un solo clasificador (*caso de uso de negocio*), es decir, no proporciona herramientas expresivas ni elementos diferenciadores. Tampoco proporciona elementos de modelado para especificar las inter-acciones ni las interacciones. Si diferenciamos las tareas del actor de negocio de las tareas del worker, y de las tareas que hacen en conjunto (inter-acciones), se requiere una expresividad distinta para representar estos tres tipos de tareas, sus roles y relaciones.

Lo que se recomienda es partir de la especificación del CU de negocio, en este caso, el especificado en la tabla 2, y de la especificación de CU de sistema (figura 6) y aplicar el proceso de clasificación para la TxA, descrito en la sección 3.2, para evaluar excesos y carencias de la actividad, además, prestando especial atención en las necesidades e intenciones de los usuarios, pues es ahí donde se encuentra el factor humano, elemento esencial en HCI (tabla 4). Por cuestiones de espacio, solo se muestra una parte de la clasificación para el caso de estudio (las abreviaturas utilizadas se aclaran en la tabla 5). A partir de esta clasificación se realizan las recomendaciones correspondientes (tabla 5), a fin de tener en cuenta en los modelos de diálogo y navegación en el diseño de las interfaces de usuario según el escenario sobre el que se hará la implementación.

La aplicación de la estructura taxonómica propuesta resulta en la misma especificación de los CU pero hecha sobre una clasificación distinta. La clasificación de la actividad en “intenciones del usuario”, “interacciones” y “acciones del sistema”, al

Tabla 3. Clasificación de la actividad para el CU, tabla 1

<i>Proceso: Inscripción</i>					
<i>Negocio</i>			<i>Sistema</i>		
<i>Aspirante</i>	<i>Inter-Acción</i>	<i>Registrador</i>	<i>Registrador</i>	<i>Interacción</i>	<i>Sistema</i>
Paso 1	Paso 2	Paso 3			
				Paso 4	
				Paso 5	
					Paso 6
					Paso 7
					Paso 8
					Paso 9
				Paso 9	
	Paso 10				
Paso 11				Paso 12	
	Paso 13				

Fuente: elaboración propia

Tabla 4. Clasificación de la actividad para el CU, tabla 2

<i>Negocio</i>			<i>Sistema</i>		
<i>Cliente: Aspirante</i>	<i>Inter-Acción</i>	<i>Worker: Registrador</i>	<i>Usuario</i>	<i>Interacción</i>	<i>Sistema</i>
Elegir Programa de Estudios.			Elegir Programa de Estudios		
Organizar información de la inscripción.			Organizar información de la inscripción.		
	El aspirante toma copia del formato UI13 y la diligencia.			El sistema muestra en pantalla formulario de inscripción. P11.	
				El Usuario diligencia formato UI13	
Verificar que la información del formulario de inscripción diligenciada sea correcta.		Verificar que la información diligenciada por el aspirante corresponda con los campos permitidos.	Verificar que la información del formulario de inscripción diligenciada sea correcta.		Validar que la información diligenciada corresponda con formato y tipos de datos
		FBN1		FBS1	AWS1
		El registrador chequea si el aspirante ya existe en el Archivo de Estudiantes y determina que el aspirante no existe en el Archivo.			El sistema chequea si el aspirante ya existe en el Archivo de Estudiantes y determina que el aspirante no existe en el Archivo.
		AWN1. FBN2.			AWS2

Fuente: elaboración propia

Tabla 5. Recomendaciones al CU “inscribirse en una universidad”
(aspectos relacionados al HCI)

<i>Atributo</i>	<i>Recomendación</i>
Feedback de Negocio (FBN)	FBN1: Indicar al aspirante si el formulario está bien diligenciado.
	FBN2: Se debe indicar al aspirante si existe en el archivo.
Feedback de Sistema (FBS)	FBS1: Se debe indicar el paso a seguir por parte del usuario o sistema luego de que éste ingresa la información, es decir, cómo sigue la interacción (presionar un botón, aceptar en una opción de pantalla..).
Awareness de Negocio (AWN)	AWN1: Se debe indicar al usuario que se está chequeando su existencia en Archivo.
Awareness de Sistema (AWS)	AWS 1: Se debe indicar al usuario que se está validando la información diligenciada.
	AWS 2: Se debe indicar al usuario que se está chequeando su existencia en Archivo.
Patrones de Interacción (PI)	PI1: Para actividades en las que se requiere ingresar información, se sugiere definir un conjunto de “Patrones de Interacción”. Para este caso, un patrón para “Ingresar Información”.

Fuente: elaboración propia

nivel de sistema, permite llevar a cabo un proceso de integración con el modelado de tareas interactivas mediante la notación CTT [19], que ha sido ampliamente difundida en la comunidad de HCI y cuyos modelos son completamente integrables con los casos de uso de UML [10].

De igual forma, la clasificación mediante la TxA permite identificar carencias en la definición de *patrones de interacción*, *tareas de feedback*, *tareas de awareness*, a partir de la especificación de la funcionalidad de un sistema, lo cual permite incorporar aspectos del diseño de la interacción y de interfaces de usuario, desde el inicio del proceso de desarrollo software. Esta incorporación aporta positivamente en los niveles de usabilidad del sistema y permite complementar la especificación del CU analizado, para mejorar la HCI.

CONCLUSIONES

La definición de una estructura taxonómica de la actividad permite evaluar la especificación de un sistema interactivo, partiendo de la práctica de separar los múltiples niveles de abstracción y de granularidad. De esta manera, la estructura taxonómica propuesta puede ser útil como un patrón de comparación o referencia para fundamentar las discusiones acerca de la efectividad, por ejemplo, de los casos de uso o del modelado esencial [20].

La utilidad y eficacia de las herramientas de modelado y especificación de sistemas interactivos, por ejemplo, los casos de uso, se deben juzgar teniendo como base

una clasificación de la actividad, pues se debe evaluar qué información se captura, en qué nivel de abstracción, siempre teniendo en cuenta el contexto y los objetivos del diseño.

En este trabajo nos hemos enfocado en el aspecto funcional, pero estamos conscientes de que para cada aspecto (colaboración, interfaz de usuario) que se desee tener en cuenta en el desarrollo de un sistema interactivo, se percibe la actividad de manera distinta.

El conjunto de clasificadores que proporciona la estructura taxonómica de la actividad se convierte en una base para la definición de elementos de modelado diferenciadores y posteriormente para la definición e implementación de herramientas que permitan la instrumentación y transformación de modelos a **código ejecutable**. Es necesario entonces definir qué otros clasificadores se deberían incluir en la propuesta.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Jacobson, I., I. Spence, and K. Bittner *Use-Case 2.0. The Guide to Succeeding with Use Cases*. 2011.
- [2] Nunes, D. N. J., *Object Modeling for User-Centered Development and User Interface Design: The Wisdom Approach*. 2001, Universidade da Madeira: Funchal.
- [3] Lozano, M.D., *Entorno metodológico orientado a objetos para la especificación y desarrollo de interfaces de usuario*. 2001, Universidad Politecnica de Valencia.
- [4] Giraldo, W. J., *Marco de desarrollo de sistemas software interactivos basado en la integración de procesos y notaciones*, Escuela Superior de Informática de Ciudad Real. 2010, Universidad de Castilla-La Mancha: Ciudad Real. p. 370.
- [5] Ng, P.-W., "Effective Business Modeling with UML: Describing Business Use Case and Realizations". *The Rational Edge*, 2002.
- [6] Gottesdiener, E., "Top Ten Ways Project Teams Misuse Use Cases -- and How to Correct Them", *Rational Edge*. 2002, Rational Software.
- [7] Constantine, L. L. and L. A. D. Lockwood, "Structure and style in use cases for user interface design", in *Object modeling and user interface design: designing interactive systems*. 2001, Addison-Wesley. pp 245-279.
- [8] Cockburn, A., *Writing Effective Use Cases*. Addison-Wesley. 2000.
- [9] Villegas, M. L., C. A. Collazos, W.J. Giraldo and J.M. González. "Activity Theory as a Framework for Activity Taxonomy in HCI". *IEEE Latin America Transactions*, Vol. 14, No. 2. 2016.
- [10] Paternò, F. "Towards a UML for Interactive Systems". in 8th International Conference on Engineering for Human-Computer Interaction. 2001. Springer.

- [11] Langlands, M. and C. Edwards, “Business vs. System Use Cases”. Disponible en: <http://www.agileea.com/Whitepapers/All3%20Business%20vs%20System%20Use%20Cases%20v1%209.pdf>. 2009.
- [12] Kramer, J. and O. Hazzan, “The Role of Abstraction in Software Engineering”, 28th International Conference on Software Engineering (ICSE) 2006, ACM: Shanghai, China.
- [13] Zachman, J. A., “The Zachman Framework for Enterprise Architecture - The Enterprise Ontology”. Disponible en: www.zachman.com. 2007-2011.
- [14] Liskov, B. and S. Zilles, “Specification techniques for data abstractions”, *Proceedings of the international conference on Reliable software*. 1975, ACM.
- [15] Giraldo, W. J., M. L. Villegas, and C.A. Collazos. Incorporación de HCI: Validación de la Usabilidad en Casos de Uso mediante la Taxonomía de la Actividad. in Computing Colombian Conference (10CCC), 2015 9th. 2015.
- [16] Iqbal, R., A. James, and R. Gatward, “Designing with ethnography: An integrative approach to CSCW design”. *Advanced Engineering Informatics*, 2005. 19(2): pp 81-92.
- [17] Molina, A. I. “CIAM: A methodology for the development of groupware user interfaces”. *Journal of Universal Computer Science(JUCS)*, 2008. 14(9).
- [18] Ambler, S.W. “System Use Cases: An Agile Introduction”. Disponible en:<http://agilemodeling.com/artifacts/systemUseCase.htm>. 2013-2014.
- [19] Paternò, F.: ‘*ConcurTaskTrees: An Engineered Notation for Task Models*’. Lawrence Erlbaum Associates, 2004. pp. 483-501.
- [20] Constantine, L. L., “Essential modeling: use cases for user interfaces”. *Interactions ACM*, 1995. 2(2): pp 34-46.