

Metamorfosis: Un Marco para el Análisis de Requisitos Funcionales*

Isabel Díaz^{1,2}, Juan Sánchez² y Oscar Pastor²

¹ Universidad Central de Venezuela – Laboratorio TOOLS – Escuela de Computación

² Universidad Politécnica de Valencia – Dpto. de Sistemas Informáticos y Computación

{idiaz|jsanchez|opastor}@dsic.upv.es

Resumen. Este artículo describe el Modelo de Realización de Metamorfosis. El propósito principal de este marco es proporcionar la base conceptual necesaria para el establecimiento de vínculos persistentes y controlables entre la especificación de los requisitos funcionales del sistema (expresado como un Modelo de Casos de Uso) y el producto de su análisis (representado a través de un Modelo de Objetos y un Modelo de Interacción). Metamorfosis permite la deducción automática de los modelos del análisis a partir del texto de los casos de uso. Con este objetivo, el Modelo de Casos de Uso ha sido enriquecido para hacer posible la representación de la información sintáctica y semántica de las acciones de los casos de uso. La utilización de roles para facilitar la comprensión de las representaciones y potenciar la genericidad, es otra característica de Metamorfosis.

1. Introducción

La descripción de las vistas funcionales, estructurales y de comportamiento de un sistema son actividades de decisiva importancia para la especificación y análisis de requisitos [1]. Los tradicionales métodos de desarrollo de sistemas orientados a objeto (MOOs) como, por ejemplo, OOSE (Object-Oriented Software Engineering), RUP (Rational Unified Process) y Fusion, utilizan: casos de uso para describir los requisitos funcionales; diagramas de clase para describir la estructura estática del sistema; y diagramas de secuencia para especificar el comportamiento del sistema [2,3,4]. Desafortunadamente, estos métodos proporcionan pocas guías para mantener la consistencia entre las diferentes vistas o entre los distintos modelos, existiendo una escasa integración entre ellos.

Una de las posibles causas de este problema es la forma cómo los MOOs conciben el proceso de análisis de requisitos a partir de la especificación de los mismos. Una vez que el Modelo de Casos de Uso ha sido desarrollado, se destina gran parte del esfuerzo a la construcción del Modelo de Objetos. Luego, esta representación es utilizada para construir el Modelo de Interacción con el casi exclusivo fin de identificar las operaciones de las clases que participan en el Modelo de Objetos previamente desarrollado. El Modelo de Interacción es subutilizado y frecuentemente desestimado [5]. Por

* Este trabajo ha sido realizado gracias al apoyo de: (i) el Proyecto DESTINO del MEC-España, Nº TIN2004-03534, cofinanciado por FEDER; y (ii) el Consejo de Desarrollo Científico y Humanístico de la Universidad Central de Venezuela (CDCH/UCV).

otra parte, pocos MOOs promueven el establecimiento de vínculos entre estos modelos con el propósito de facilitar su construcción, garantizar la consistencia de los mismos y determinar oportunamente cómo afectan a unos modelos las modificaciones que se efectúan en otro.

En este trabajo se describe un marco ideado para estrechar la brecha entre la especificación y el análisis de los requisitos funcionales e intentar solventar los problemas que esta brecha ocasiona. El marco, denominado Metamorfosis, fue concebido para ser integrado en ambientes de producción automática de software. Para su descripción se ha utilizado la sintaxis y la semántica del Unified Modeling Language (UML) [6]. Metamorfosis está basado en: (i) la definición de *roles*, para potenciar la semántica de los modelos y promover la genericidad [7,8]; (ii) la *transformación automática* de un modelo para la obtención de otro [9]; y (iii) la aplicación de *técnicas de procesamiento de lenguaje natural* para el reconocimiento de elementos de modelado a partir del texto de los casos de uso [10].

Este artículo ha sido organizado en seis secciones. La Sección 2 describe el Modelo de Realización de Metamorfosis y cómo este marco plantea el análisis del Modelo de Casos de Uso del sistema. La Sección 3 presenta el contexto de realización de una acción como parte fundamental de un caso de uso; contiene tres subsecciones que describen, respectivamente: la especificación de una acción, cómo se concibe su análisis y cómo se realiza de forma automática utilizando un conjunto de reglas predefinidas. La Sección 4 describe brevemente algunos trabajos relacionados. En las dos últimas secciones se exponen las conclusiones, los trabajos futuros y las referencias.

2. Modelo de Realización

La Figura 1 muestra el Modelo de Realización de Metamorfosis. Cada columna de esta representación es un contexto para el análisis de la especificación del comportamiento: *sistema*, *caso de uso* y *acción*. Cada contexto delimita el ámbito al cual pertenece la información descrita en los modelos que éste contiene. Los modelos de contextos de realización distintos están vinculados entre sí por dos tipos de relaciones de dependencia [6]. Las relaciones de integración y disgregación permiten vincular fragmentos de un mismo tipo de modelo. Pueden ser establecidas siempre que el fragmento cliente incluya información del fragmento que cumple el rol de servidor.

- La *relación de integración* representa la adecuada combinación de la información del contexto inmediato inferior, garantizando la consistencia y completitud de los modelos obtenidos. Esto significa que toda la información obtenida en un contexto debe contener aquella que pertenece a los modelos del contexto inmediato inferior. Además, esto debe hacerse resolviendo las ambigüedades que se pudieran generar como consecuencia del proceso de integración.
- La *relación de disgregación* expresa la conveniente separación en partes de un modelo. Cada parte o fragmento es definida como una unidad que puede ser analizada independientemente de las otras.

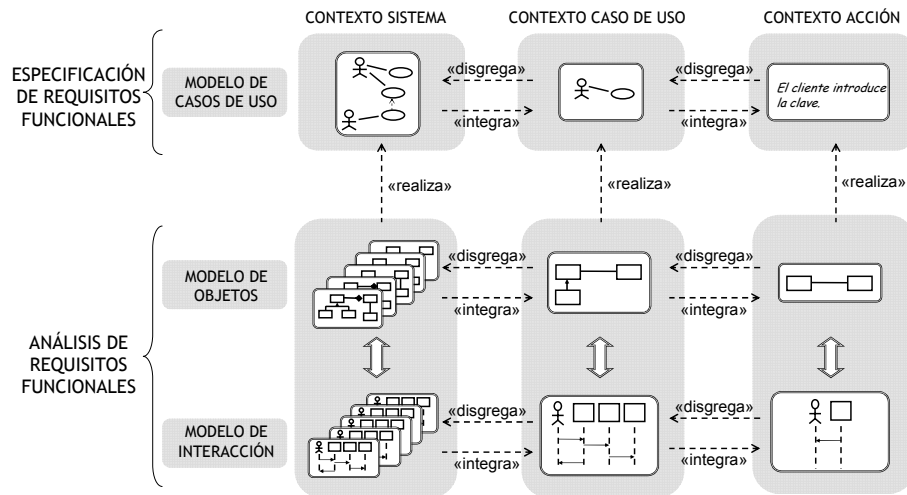


Fig. 1. Modelo de Realización de Metamorfosis

Las *relaciones de realización*, definidas entre los modelos de cada contexto, son relaciones de dependencia por abstracción [6]. Permiten vincular elementos que representan un mismo concepto que ha sido expresado a distintos niveles de abstracción. En particular, el Modelo de Realización de Metamorfosis muestra la relación establecida entre la especificación y el análisis del comportamiento del sistema en desarrollo.

En el *contexto del sistema*, la especificación de todos sus requisitos funcionales es expresada a través del Modelo de Casos de Uso. La realización de este modelo es representado mediante el Modelo de Objetos y el Modelo de Interacción. En estos modelos se describen todas las clases de objetos del sistema, sus propiedades y restricciones, y también la forma cómo estas clases deben interactuar entre sí para dotar al sistema del comportamiento especificado en los casos de uso. Este contexto integra toda la información a nivel de especificación y análisis de los otros contextos definidos.

Cada caso de uso define un contexto de análisis. El *contexto de un caso de uso* sólo describe una parte del comportamiento completo del sistema. Así, el resultado de su análisis muestra únicamente la información estática y dinámica necesaria para dotar al sistema del comportamiento especificado en este fragmento. En Metamorfosis, un caso de uso es una especificación en lenguaje natural de la secuencia completa de acciones que debe realizar el sistema cuando interactúa con los actores con el propósito de lograr cierto objetivo. Cada una de estas acciones, obtenidas por disgregación del caso de uso, define un contexto de realización. Los modelos del análisis de un caso de uso integran la información descrita en cada una de sus acciones.

3. Contexto de Realización de una Acción

El Modelo de Realización de Metamorfosis está basado en la integración de información estructural y dinámica deducida del análisis de una acción. Esta sección está dedicada a describir el contexto de realización de una acción. Para esto, es necesario conocer cómo puede ser especificada así como también cuáles son las características de los elementos que participan en los fragmentos que resultan del análisis de la misma.

3.1 Especificación de una acción

A nivel de especificación, el contexto de realización de una acción muestra: la *comunicación* establecida entre un actor y el sistema para el intercambio de información; o bien, la *actuación interna* del sistema en respuesta a una comunicación establecida con el actor. Además de las acciones de comunicación y de actuación interna, en un caso de uso se pueden distinguir las acciones *especiales*. Estas acciones describen la extensión o inclusión de casos de uso así como también el condicionamiento o iteración de acciones.

Una acción puede ser: unidireccional y atómica. Se dice que una acción es *unidireccional* si se refiere a alguna de las siguientes situaciones: (i) un actor se comunica con el sistema, introduciéndole información; (ii) el sistema se comunica con el actor para suministrarle información; o (iii) el sistema ejecuta una acción (sin la participación directa del actor). La Figura 2 muestra todas las posibles direcciones de una acción. Una acción es atómica si no puede descomponerse en dos o más acciones unidireccionales (en caso contrario, se diría que la acción es compleja). Por otra parte, la especificación de una acción puede ser estudiada desde dos perspectivas distintas pero complementarias: la perspectiva sintáctica y la perspectiva semántica.

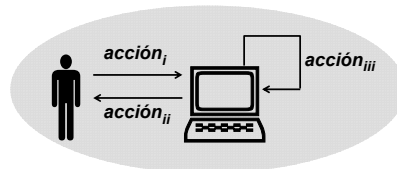


Fig. 2. Unidireccionalidad de una Acción

Desde la *perspectiva sintáctica* o *gramatical*, una acción es una oración del texto del caso de uso. Una *oración simple* representa una acción unidireccional de comunicación o de actuación del sistema. Las *oraciones especiales* corresponden a acciones especiales. Una oración simple es una secuencia de palabras asociadas a una categoría sintáctica (sustantivo, verbo, adjetivo, etc.). Estas palabras pueden ser agrupadas para formar estructuras sintagmáticas que cumplen determinada función gramatical en la oración [11]. Con el propósito de evitar las ambigüedades generadas por el uso del lenguaje natural, la oración que representa una acción unidireccional debe: (i) ser redactada de forma declarativa, afirmativa y activa; (ii) contener un solo

sujeto y un único verbo principal; (iii) ser transitiva para asegurar la presencia del objeto directo. La Figura 3 muestra la vista sintáctica de una acción [12].

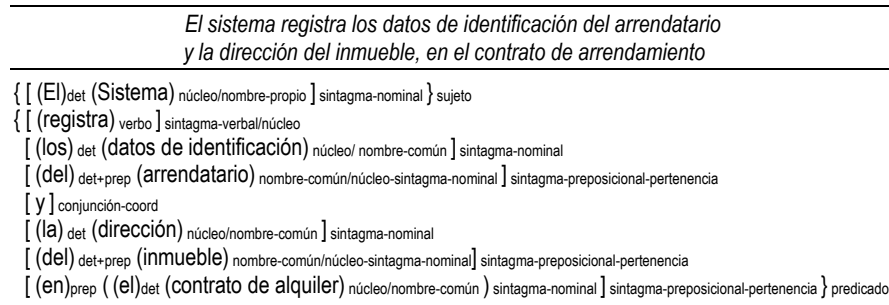


Fig. 3. Perspectiva Sintáctica de una Acción

Una oración simple representa una acción compleja si puede ser descompuesta en dos o más cláusulas (oraciones incrustadas). Cada cláusula es también una oración simple que mantiene la unidireccionalidad de aquella que la contiene (oración matriz). De esta forma, si la oración matriz representa una acción unidireccional de comunicación, las oraciones incrustadas también representarán este tipo de acciones. Desde el punto de vista gramatical, una oración matriz que tiene dos o más cláusulas es una oración compuesta por coordinación que tiene un único sujeto. Si de una acción es posible deducir únicamente una cláusula, debe ocurrir que la acción y la cláusula son la misma. Nótese que la oración de la Figura 3 representa una acción compleja que puede ser descompuesta en dos acciones unidireccionales atómicas de actuación interna, cada una de las cuales es una cláusula: (i) *El sistema registra los datos de identificación del arrendatario en el contrato de arrendamiento*; (ii) *El sistema registra la dirección del inmueble en el contrato de arrendamiento*.

Desde la *perspectiva semántica*, una acción es expresada como una relación establecida entre roles semánticos [8]. Un rol semántico describe una función abstracta desempeñada por un elemento que participa en una acción. Esta función abstracta es definida independientemente de las distintas estructuras sintácticas que puede adquirir una oración. De esta forma, los roles semánticos permiten representar de forma genérica acciones, sin depender del idioma utilizado para escribir los casos de uso ni de los diversos recursos gramaticales que ofrece un idioma para expresar una misma acción.

Los roles semánticos utilizados en Metamorfosis pueden ser: (i) *básicos*, si participan siempre en cualquier acción como, por ejemplo, el rol agente (qué/quién realiza la acción) y el objeto (qué/quién recibe o sufre la acción del verbo); (ii) *secundarios*, si no son requeridos por la acción verbal como, por ejemplo, el rol de ubicación (destino del objeto cuando la acción se realiza); el rol de propietario (qué/quién posee algo) y el rol de pertenencia (lo poseído).

Para caracterizar una acción utilizando roles semánticos, Metamorfosis utiliza tres recursos: (i) una *fórmula lógica* cuya evaluación permite determinar si una acción tiene o no las características semánticas establecidas por los roles; (ii) un *diagrama de clases* que representa gráficamente los roles que participan en la acción, las relaciones entre éstos y sus restricciones; (iii) una *descripción* en lenguaje natural de la acción. La Figura 4 presenta un ejemplo de cómo estos tres recursos se complementan entre sí para dar información sobre un tipo particular de acciones utilizando roles semánticos.

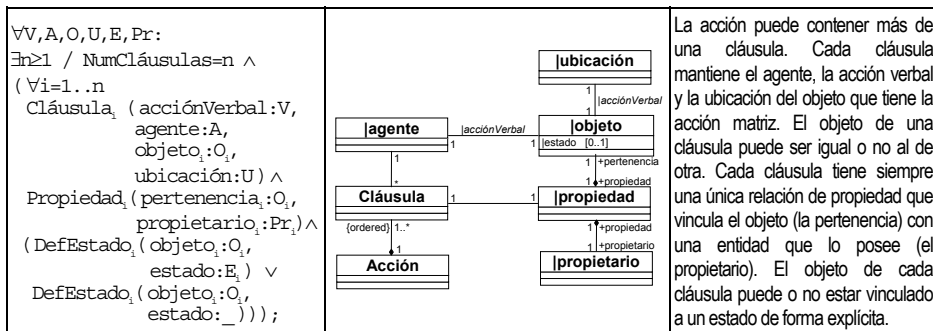


Fig. 4A. Fórmula

Fig. 4B. Representación Gráfica

Fig. 4C. Descripción

Fig. 4. Perspectiva Semántica de una Acción

Los términos de la fórmula de una acción pueden ser variables que representan roles semánticos, constantes auxiliares y/o funciones aplicables a los términos de la fórmula. La fórmula de la Figura 4A describe una acción a través de los roles: agente, acciónVerbal, objeto, ubicación, pertenencia, propietario y estado. Cláusula, Propiedad y DefEstado son funciones booleanas que permiten determinar si una acción tiene o no los roles semánticos utilizados como argumentos. El uso del símbolo "_" indica la ausencia del elemento. Así, el cambio del estado del objeto después de la ejecución de la acción puede o no haber sido expresado de forma explícita en la oración. La aplicación de la operación NumCláusula permite calcular la cantidad de cláusulas que se pueden obtener a partir de la acción matriz. Un ejemplo de cómo evaluar la fórmula de la Figura 4A se presenta en la Figura 5. Allí se demuestra que la acción de la Figura 3 satisface esta fórmula.

La Figura 4B describe la fórmula de una acción utilizando un diagrama de clases. La información que aporta esta representación puede ser complementada con otros tipos de especificaciones (por ejemplo, con restricciones escritas en algún lenguaje). El diagrama muestra algunas clases estereotipadas como roles. Estas clases se pueden distinguir por formato textual o gráfico que indica el estereotipo: «rol» o |nombreRol [6,7]. Según el diagrama, la acción puede estar compuesta por una o más cláusulas. Cada cláusula tiene un único agente relacionado con un único objeto mediante la relación acciónVerbal. El objeto está vinculado también a una única ubicación a través de la acciónVerbal. Por otra parte, en cada cláusula debe haberse

establecido una relación de propiedad entre un propietario y su pertenencia. El objeto de cada cláusula es, además, la pertenencia en la relación de propiedad. Además, cada objeto puede o no tener un estado explícitamente expresado en la acción.

*El sistema registra los datos de identificación del arrendatario
y la dirección del inmueble, en el contrato de arrendamiento*

```

NumCláusulas=2 ^
Cláusula1 ( acciónVerbal: registra, agente: el sistema,
              objeto1: los datos de identificación, ubicación: en el contrato de arrendamiento) ^
Propiedad1 ( pertenencia1: los datos de identificación, propietario1: del arrendatario) ^
DefEstado1 ( objeto1: los datos de identificación, estado: _ ) ^
Cláusula2 ( acciónVerbal: registra, agente: el sistema,
              objeto2: la dirección, ubicación: en el contrato de arrendamiento) ^
Propiedad2 ( pertenencia2: la dirección, propietario2: del inmueble) ^
DefEstado2 ( objeto2: la dirección, estado: _ ) ;

```

Fig. 5. Evaluación de una Fórmula

3.2 Análisis de una acción

El resultado del análisis de una acción es expresado desde dos perspectivas: la estática y la dinámica. Desde la *perspectiva estática*, el análisis es representado como un fragmento del Modelo de Objetos del sistema. Este fragmento muestra únicamente aquellas clases que son estrictamente requeridas para que el comportamiento descrito en la acción pueda ser realizado. Las propiedades, relaciones entre clases y restricciones que se presentan en este fragmento también están limitadas a los elementos que son necesarios para realizar la acción.

Para la representación del fragmento estructural de una acción, es utilizada la notación UML para los diagramas de clases [6]. Conviene hacer dos observaciones al respecto. En primer lugar, debido a que la información suministrada por el fragmento está limitada al contexto de una acción, ésta puede ser incompleta en relación a la que es expresada en el Modelo de Objetos del sistema; para evitar confusión, aquellos elementos de este fragmento que son susceptibles de modificación son colocados entre paréntesis (por ejemplo, la multiplicidad de las relaciones entre clases). En segundo lugar, la descripción genérica del fragmento se basa en el uso de roles cada uno de los cuales desempeña el papel de alguno de los elementos de modelado del fragmento. Esto permite asignarles una semántica particular.

La Figura 6 muestra un fragmento estructural genérico correspondiente a una acción. El rol |contenedor es desempeñado por una clase entidad del dominio del sistema. Adicionalmente, participan en este fragmento n clases entidad representadas por el rol |ejecutor. Para evitar recargar la representación, no se muestran todas estas clases (se aclara en una nota). Entre las clases |contenedor y |ejecutor- i se ha establecido una relación de agregación que presenta a esta

última como parte de la primera. Nótese que la multiplicidad indica que una sola instancia `|ejecutor-i` forma parte de `|contenedor`. La clase-todo `|contenedor` está relacionada con n clases-parte `|ejecutor`. La relación de agregación pudiera ser por composición. Esto depende de las características de la acción por lo que tal decisión debe ser tomada por el analista. Cada `|ejecutor-i` tiene, al menos, un atributo que es representado por $|p_i$ y una operación destinada a cambiar su valor. Este tipo de operación es desempeñada por `|actualizar(|p_i)`. Por último, `|iniciador` puede desempeñar el rol de una clase frontera, de una clase control o de ambas. Además, su participación en el fragmento estructural es opcional. Tales decisiones deben ser tomadas por el analista.

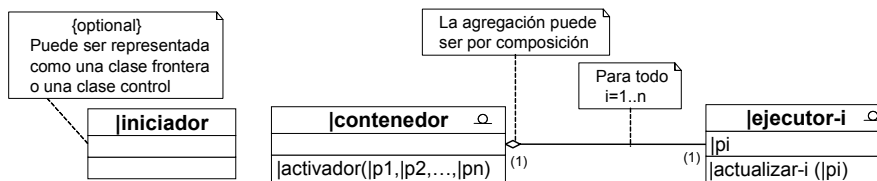


Fig. 6. Análisis de una Acción: Perspectiva Estructural

Desde la *perspectiva dinámica*, el análisis de una acción es representado como un fragmento del Modelo de Interacción del sistema. Un fragmento de interacción muestra cómo las instancias de las clases que participan en una acción intercambian información para que el comportamiento que ésta describe pueda realizarse. En términos UML, un fragmento de interacción es también una interacción y puede estar formada por una o más instancias que intercambian uno o más mensajes [6]. La representación gráfica de un fragmento de interacción es un diagrama de secuencia UML. La descripción genérica de la interacción está basada en roles que tienen como propósito agregar semántica al fragmento y facilitar su comprensión.

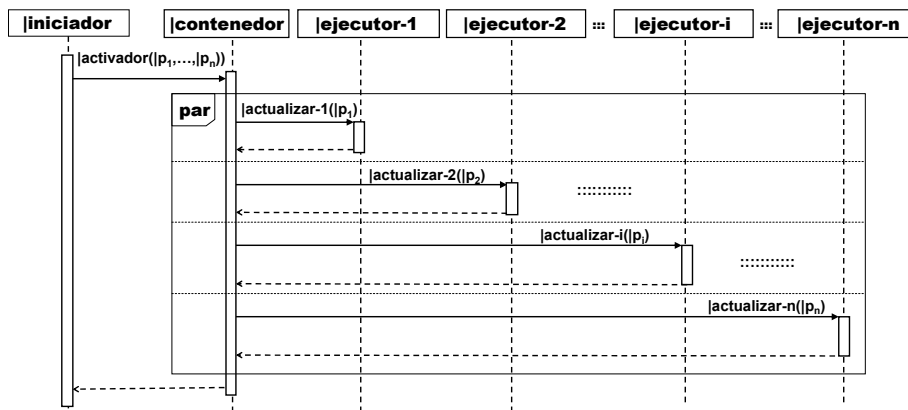


Fig. 7. Análisis de una Acción: Perspectiva Dinámica

En la Figura 7 se puede observar un ejemplo de un fragmento de interacción. La representación gráfica utiliza la notación de roles para distinguir los elementos del diagrama. La expresión $|iniciador$ identifica un rol que es desempeñado por la línea de vida que inicia la secuencia de mensajes. Esta línea de vida representa una instancia de una clase de tipo frontera o control [13]. El resto de las líneas de vida del fragmento representan clases de tipo entidad. La línea de vida que es receptora de este mensaje cumple el rol de $|contenedor$. El mensaje $|activador$ es un tipo de operación que cambia el estado de la instancia que representa la línea de vida receptora. Este mensaje tiene, al menos, n parámetros $(|p_1, |p_2, \dots, |p_n)$. La recepción del $|activador$ induce al $|contenedor$ a enviar n mensajes a n líneas de vida $|ejecutor$, respectivamente. Así, $|ejecutor-i$ es la i -ésima línea de vida responsable de ejecutar el i -ésimo mensaje $|actualizar-i$ que representa una operación que modifica el estado de la instancia. Además, cada mensaje $|actualizar-i$ tiene, al menos, un parámetro $|p_i$.

El operador de interacción Par definido por el UML indica que la línea de vida $|contenedor$ puede enviar los mensajes de forma simultánea y que, además, pueden ser ejecutados por cada instancia receptora al mismo tiempo [6]. Una vez ejecutados todos los mensajes de la corrección, $|contenedor$ envía la correspondiente respuesta a la línea de vida $|iniciador$.

3.3 Realización de una acción

La realización de una acción modela la relación establecida entre los elementos de la especificación de dicha acción y aquellos elementos que son utilizados para expresar el resultado de su análisis. Con el propósito de garantizar el establecimiento de esta relación, Metamorfosis aplica un conjunto de reglas. Las reglas de realización permiten deducir el fragmento estructural y el fragmento de interacción correspondiente a una acción. Es importante hacer notar que ambas representaciones son obtenidas utilizando las mismas reglas de realización. De esta forma, la consistencia de los modelos es inherente a la construcción de los modelos del análisis. Por otra parte, la deducción de un tipo de fragmento no depende de la deducción del otro. Aunque las reglas permiten generar simultáneamente ambos fragmentos, cada uno de éstos tiene una existencia propia por lo que pudieran construirse también separadamente y en momentos distintos.

La Figura 8 presenta las reglas de realización aplicables a acciones del tipo descrito en la Figura 4. Los fragmentos que se obtienen adquieren la forma genérica que se muestra en las Figuras 6 y 7. La especificación de estas reglas está basado en los roles utilizados en estas representaciones. Del lado izquierdo de cada regla están presentes los roles de los fragmentos. Del lado derecho, las expresiones utilizan los roles que participan en la acción y operaciones que permiten obtener información más detallada de estos roles.

iniciador	← agente
contenedor	← <Núcleo(ubicación)> ^{Norm}
ejecutor-i	← <Núcleo(proprietario-i)> ^{Norm} $\forall i=1..n$
activador	← Secuencia(<acciónVerbal> ^{Norm} , Propiedad<objeto-i, propietario-i> $\forall i=1..n$)
actualizar-i	← Secuencia(<acciónVerbal> ^{Norm} , <objeto-i> ^{Norm}) $\forall i=1..n$
p_i	← <Núcleo(objeto-i)> ^{Norm} $\forall i=1..n$

Fig. 8. Reglas de Realización

La primera de las reglas de realización de la Figura 8 indica que el rol de |iniciador de los fragmentos es desempeñado por el |agente de la acción que es analizada. El |contenedor es deducido de |ubicación. La operación Núcleo permite extraer el constituyente más importante del rol. Además, la operación Norm es responsable de expresar en su forma canónica el resultado obtenido. La línea de vida |ejecutor-i es obtenida del núcleo normalizado del |propietario-i. Cada uno de los parámetros | p_i es deducido también normalizando el núcleo de cada |objeto-i, respectivamente. El mensaje |activador es una secuencia formada por |acciónVerbal y todos los pares que relacionan una pertenencia con su propietario (|objeto-i, |propietario-i). Por último, cada operación |actualizar-i es identificada por la secuencia normalizada formada por la |acciónVerbal y el correspondiente |objeto-i.

Para aplicar las reglas de realización a una acción de un caso de uso, debe antes vincularse cada rol de la especificación con una estructura sintáctica. Para mostrar cómo Metamorfosis concreta esta aplicación, se utilizará la acción descrita desde las perspectivas sintáctica y semántica en las Figuras 3 y 5. En estas figuras es posible observar, por ejemplo, que el |agente y la |pertenencia de la acción están vinculados a sintagmas nominales mientras que, la |ubicación es un sintagma preposicional de lugar (la preposición denota lugar). Dado que la acción utilizada para ejemplificar satisface la fórmula descrita en la Figura 3, es posible aplicar las reglas de realización que se presentan en la Figura 8. Las Figuras 9 y 10 muestran, respectivamente, el fragmento estructural y el fragmento de interacción de la acción, obtenidos por realización.

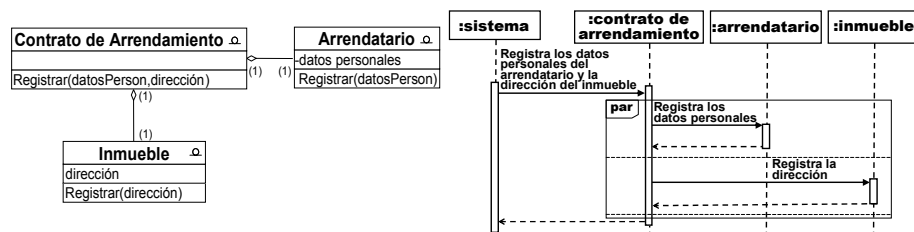


Fig. 9. Fragmento Estructural de una Acción Fig. 10. Fragmento de Interacción de una Acción

La definición de los patrones conceptuales se ha realizado en dos etapas. La primera etapa consistió en la identificación y especificación de patrones de transformación para la deducción de fragmentos de interacción de acciones. Hasta ahora se han validado 21 patrones de interacción de forma manual y automática. La estrategia de validación definida, la herramienta creada para tal fin y los resultados obtenidos pueden ser consultados en [14]. En la segunda etapa, cada patrón de interacción fue extendido con información relativa a cómo obtener el fragmento estructural de una interacción. Los patrones conceptuales especificados sólo han sido validados manualmente por modeladores expertos, utilizando sistemas académicos. Actualmente se está diseñando la estrategia de validación automática de los patrones conceptuales la cual será aplicada próximamente.

4. Trabajos Relacionados

Con el fin de reducir el amplio espectro de contribuciones realizadas para la deducción automática de modelos a partir de casos de uso, se distinguirán dos grupos: aquellas que permiten obtener el Modelo de Objetos, de las aportaciones dedicadas a la generación de Modelos de Interacción. En el primer grupo, son representativas las aproximaciones descritas en [10,15,16]. Del segundo grupo, pueden distinguirse los trabajos de Feijs y Li [17,18]. La diferencia entre las aproximaciones de ambos grupos y la propuesta que nosotros realizamos es que la deducción de los modelos de análisis no se hace de forma simultánea ni tampoco establecen mecanismos para garantizar la consistencia de los modelos como parte de la construcción de los mismos.

5. Conclusiones y Trabajos Futuros

En este artículo se ha descrito el Modelo de Realización de Metamorfosis. Este modelo muestra cómo se concibe la transformación automática del Modelo de Casos de Uso de un sistema para obtener el Modelo de Objetos y el Modelo de Interacción del mismo. El Modelo de Realización define tres contextos de realización (sistema, caso de uso y acción). Estos contextos se vinculan entre sí a través de las relaciones de integración y disgregación. Utilizando la relación de disgregación, la especificación del comportamiento del sistema es fraccionada en casos de uso y, cada uno de éstos, en acciones. A nivel de análisis, cada parte del comportamiento es presentada como un fragmento estructural y un fragmento de interacción. La relación de integración permite combinar convenientemente los modelos de un contexto hasta obtener los modelos de otro.

En el contexto de realización de una acción, la especificación muestra su perspectiva sintáctica y semántica. Para que esta especificación pueda ser transformada en el fragmento estructural y en el fragmento de interacción correspondientes, la acción debe satisfacer una fórmula. Si esto es así, Metamorfosis utiliza un mismo conjunto de reglas de realización para deducir los fragmentos del análisis de forma automática. Estas reglas permiten garantizar la consistencia de la información representada en cada uno de los fragmentos sin que, por esto, la existencia de uno dependa de la del otro. El

uso de roles es fundamental en Metamorfosis. Éstos permiten describir de forma genérica los artefactos que describen una acción tanto a nivel de especificación como a nivel de análisis. Además, el uso de roles independiza el Modelo de Realización del idioma utilizado para redactar los casos de uso. Actualmente, nuestro trabajo está orientado a establecer la validez y completitud de las reglas de realización de acciones definidas hasta ahora. Además, se adelanta el diseño de una estrategia de integración de los resultados obtenidos para cada acción con el fin de obtener los fragmentos del análisis correspondientes a los casos de uso y al sistema completo.

6. Referencias

- [1] Nuseibeh B., Easterbrook S. Requirements Engineering: A Roadmap. Proceedings of the 22nd Conference on Software Engineering (ICSE 2000), Conference on The Future of Software Engineering, pp. 37-46. ACM Press.
- [2] Jacobson I., Christerson M., Jonsson P., Övergaard G.: Object-Oriented Software Engineering. A Use Case Driven Approach, Addison-Wesley, 1992.
- [3] Rational Software Corporation: The Rational Unified Process. <http://www.rational.com/rup>.
- [4] Coleman D., Arnold P., Bodoff S., Dollin Ch., Gichrist H.: Object-Oriented Development. The Fusion Method. Prentice-Hall, Object-Oriented Series, 1994.
- [5] Song I.-Y.: Developing Sequence Diagrams in UML. Proceedings of the 20th International Conference on Conceptual Modeling (ER'01), LNCS, Springer-Verlag, 2001.
- [6] Object Management Group: Unified Modeling Language: Superstructure Specification, Version 2.0. August 2003, <http://www.omg.org/uml>.
- [7] France R., Kim D., Ghosh S., Song E.: A UML-Based Pattern Specification Technique. IEEE Transactions on Software Engineering, Vol. 30, N° 3, March 2004, pp. 193- 206.
- [8] Gildea D., Jurafsky D.: Automatic Labeling of Semantic Roles. Computational Linguistics, 28(3): 245-280, 2002.
- [9] Object Management Group: MDA Guide. Ver. 1.01. Jun 03. <http://www.omg.org/uml>.
- [10] Burg J.F.M., Van de Riet R.P.: Analyzing Informal Requirements Specifications: A First Step towards Conceptual Modeling. Proceedings of the 2nd International Workshop on Applications of Natural Language to Information Systems (NLDB'96), 1996.
- [11] Díaz I., Moreno L., Pastor O., Matteo A.: Interaction Transformation Patterns based on Semantic Roles. Proceedings of the 10th Inter. Conference on Applications of Natural Language to Information Systems (NLDB'05), LNCS Springer-Verlag, Junio 2005.
- [12] Díaz I., Losavio F., Matteo A., Pastor O.: A Specification Pattern for Use Cases. Information & Management, Vol. 41/8 (2004), pp. 961-975, Elsevier Science B.V.
- [13] Larman C.: Applying UML and Patterns: an Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd ed). USA: Prentice-Hall, 2004.
- [14] Díaz I., Moreno L., Fuentes, I., Pastor O.: Integrating Natural Language Techniques in OO-Method. Proceedings of the 6th Inter. Conference on Intelligent Text Processing and Computational Linguistics (CICLing'05), LNCS Springer-Verlag, Febrero 2005.
- [15] Juristo N., Moreno A.M., López M.: How to use Linguistic Instruments for Object-Oriented Analysis. IEEE Software 17(3), 2000. Pp. 80-89.
- [16] Kop Ch., Mayr H.: Mapping Functional Requirements: from Natural Language to Conceptual Schemata. Proceedings of the 6th IASTED Inter. Conference on Software Engineering and Applications (SEA). Cambridge, 2002.
- [17] Feijs L.M.G.: Natural Language and Message Sequence Chart Representation of Use Cases. Information and Software Technology 42 (2000). Pp. 633-647
- [18] Li L. Translating Use Cases to Sequence Diagrams. Proceedings of the Fifteenth IEEE International Conference on Automated Software Engineering (ASE 2000).