

INFLUENCIA DE LAS RELACIONES ENTRE ELEMENTOS SOFTWARE REUTILIZABLES EN LA GENERACIÓN DE MECANOS*

Francisco José García Peñalvo[✉]
e-mail: fgarcia@ubu.es

José Manuel Marqués Corral[✉]
e-mail: jmmc@infor.uva.es

Miguel Ángel Laguna[✉]
e-mail: mlaguna@infor.uva.es

Jesús Manuel Maudes Raedo[✉]
e-mail: jmaudes@ubu.es

Resumen

Los mecanos, elementos software reutilizables formados por assets clasificados en diferentes niveles de abstracción y por las relaciones existentes entre ellos, definen un modelo de reutilización en el que se aumenta el nivel de abstracción del proceso de reutilización dando soporte a los assets obtenidos durante todo el ciclo de vida del software.

Un aspecto importante es la creación de estas estructuras complejas de reutilización, para lo cual se cuenta con un enfoque mixto que combina los dos enfoques clásicos de reutilización, generación y composición, para obtener los mecanos mediante una generación basada en la composición de assets. Este proceso de generación de mecanos mediante composición puede realizarse de forma manual guiada por un responsable, produciéndose una serie de mecanos “persistentes” que serán almacenados en un repositorio, o de forma automática aplicando un algoritmo de generación sobre los assets almacenados en el repositorio, consiguiendo mecanos en tiempo de reutilización, que podrán ser posteriormente almacenados en el repositorio como mecanos persistentes.

En todo el proceso de construcción de mecanos las relaciones semánticas entre los assets componentes desempeñan un papel protagonista, determinando la estructura de los mecanos, así como la base para la generación automática de los mismos.

En este trabajo se estudian las relaciones semánticas que dan soporte a los mecanos y a su generación, haciendo especial hincapié en aquellas características de las relaciones que tienen una influencia directa en este proceso.

Palabras Clave

Mecano, relación estructural, relación semántica, reutilización de software, estructura compleja de reutilización, asset.

1. Introducción

Dentro de la reutilización del software pueden distinguirse dos tipos de tecnologías: las tecnologías de generación y las tecnologías de composición. Las primeras abordan la reutilización del software desde un punto de vista generativo o de procesador reutilizable, mientras las segundas presentan un enfoque de bloques de construcción.

* Este trabajo ha sido parcialmente financiado por el proyecto CICYT TIC97-0593-C05-05.

[✉] Departamento de Ingeniería Electromecánica y Civil. Área de Lenguajes y Sistemas Informáticos. Escuela Universitaria Politécnica, Universidad de Burgos. Av. General Vigón S/N, 09006 Burgos (España).

[✉] Departamento de Informática. Edificio de Tecnologías de la Información y las Telecomunicaciones. Universidad de Valladolid. Paseo del Cementerio S/N, 47011 Valladolid (España).

Tomando como base la estructura de reutilización de granularidad gruesa definida en diferentes niveles de abstracción, denominada mecano [10, 11], se define un modelo de reutilización mixto composición-generación.

El enfoque compositivo se utiliza en el desarrollo de elementos software reutilizables, siendo una actividad propia del desarrollo para reutilización, que parte de un espacio de reutilización formado por assets clasificados en diferentes niveles de abstracción y por sus interrelaciones. Como resultado se obtienen unas estructuras reutilizables de granularidad gruesa, que se almacenan en un repositorio y que se denominan mecanos.

Sin embargo, limitar el proceso de generación de mecanos a este único enfoque daría como resultado una estructura de reutilización estática, en el sentido de que sólo se podrían recuperar los mecanos almacenados en el repositorio. Aumentar el potencial de reutilización ofertado por el repositorio pasa por introducir un enfoque generativo en el modelo de reutilización que permita generar estructuras complejas de reutilización, mecanos, de acuerdo a las necesidades específicas del reutilizador dentro de las que proporciona el suministrador del repositorio. Esto es, generar nuevos mecanos a partir de los assets del repositorio que se ajusten a unos requisitos expresados por el desarrollador con reutilización.

El proceso generativo está soportado por el carácter de dependencia de los assets seleccionados con el resto de los assets del espacio de reutilización, dependencia que se ve reflejada en las relaciones semánticas entre los assets.

Según esto, el desarrollo de mecanos para su almacenamiento en el repositorio (*mecanos persistentes*) está más relacionado con las actividades propias del desarrollo para reutilización: identificación, comprensión, clasificación y almacenamiento en un repositorio. Mientras que los mecanos generados en tiempo de reutilización son más acordes con las actividades del desarrollo con reutilización: selección de assets, un proceso de generación automática del mecano navegando a través de las relaciones existentes entre los assets seleccionados, y por último el estudio, filtrado, adaptación e integración del mecano resultante. El proceso finaliza, bien cerrando el ciclo y realimentando el repositorio con el mecano generado, o bien desechando el mecano generado.

En este trabajo se realiza un estudio del papel y de la influencia que las relaciones semánticas tienen en el proceso de generación de mecanos. Para ello, en la sección dos se hace un repaso de las relaciones semánticas entre elementos software reutilizables. En la tercera sección se ofrecen las soluciones adoptadas en el espacio de reutilización donde se definen los mecanos,

haciendo especial hincapié en la justificación de un conjunto reducido de relaciones estructurales, que sirven como base a la construcción de los mecanos, así como a sus características. Para terminar, en la cuarta sección se presentan las conclusiones de este trabajo.

2. Relaciones entre elementos software reutilizables

Las relaciones entre los elementos software reutilizables, assets, son fundamentales en los mecanos. Pero, antes de entrar en la solución adoptada en la definición del espacio de reutilización de los mecanos, se va a hacer un repaso de la influencia de las relaciones semánticas entre assets en otros importantes trabajos sobre reutilización del software.

En el proyecto EEC-SPRIT II ITHACA (Integrated Toolkit for Highly Advanced Computers Applications) [2] se presenta un repositorio, que recibe el nombre de SIB (Software Information Base) [6], donde se almacenan las descripciones de los objetos software y sus relaciones semánticas. Las relaciones semánticas del modelo ITHACA se clasifican en tres categorías: *relaciones semánticas estructurales generales* (mecanismos básicos de modelado: atribución, clasificación y generalización), *relaciones semántica estructurales especiales* (agregación, correspondencia, similitud y especificidad) y *asociaciones*.

Dentro de las relaciones establecidas por ITHACA cabe hacer una mención especial a un tipo de asociación de granularidad gruesa, denominada *Application Frame*, que representa familias de sistemas software, incluyendo descripciones de implementación y opcionalmente de diseño y requisitos.

Las relaciones establecidas por ITHACA suponen un punto de influencia importante para los mecanos, especialmente en lo referente a relacionar assets de diferentes niveles de abstracción, cambiando la opcionalidad de ITHACA por obligatoriedad en los mecanos. No obstante, cabe destacar una clara diferencia entre ITHACA y los mecanos, mientras que ITHACA presenta relaciones estructurales que por un lado relacionan a una entidad con sus componentes estructurales y por otro relacionan entidades, en los mecanos sólo interesa definir las relaciones entre entidades (assets) dejando el otro tipo de relaciones al modelo objeto subyacente.

Otra importante referencia la constituye el Reusable Software Research Group (RSRG) de la Ohio State University. Este grupo ha definido un pequeño conjunto de relaciones semánticas independientes de cualquier lenguaje entre componentes [8], encuadrándose dentro del modelo de reutilización 3C [16], que sirve de marco de referencia conceptual para

subsistemas software reutilizables. Las relaciones identificadas por el RSRG intentan describir las dependencias entre componentes a un nivel conceptual, de manera que las relaciones ofrezcan a los programadores y encargados del mantenimiento una información precisa sobre la forma en que los componentes pueden y deben ser usados. Del conjunto de relaciones definidas, las tres relaciones más generales son implementa, usa y extiende.

Bajo las ideas del RSGR se puede apreciar claramente una intención de relacionar componentes clasificados en diferentes niveles de abstracción, aunque no se pretende en ningún momento definir un componente multinivel de abstracción. Otro aspecto a destacar es que su modelo de reutilización está montado sobre su filosofía de implementación de sistemas software, donde las especificaciones no son otra cosa que tipos abstractos de datos, no cubriendo por tanto todo el ciclo de vida del software.

En este repaso por las relaciones semánticas no puede dejarse de hacer una referencia a los diferentes modelos de assets soportados por los principales repositorios EUROWARE [15], RIB [14], RBSE [9] entre otros, de cuyo estudio se pueden extrapolar las siguientes conclusiones generales:

- En la mayoría de los repositorios el concepto de relación semántica aparece (siendo inexistente en algún caso) con la única connotación de enlace entre assets.
- La semántica de las relaciones suele recaer en el nombre de las mismas, siendo totalmente genéricas y sin implementar restricción alguna.
- La connotación semántica más difundida es la de composición, buscando representar diferentes grados de granularidad en los assets.

Notables excepciones a los casos anteriormente mencionados son REBOOT [12], que establece un conjunto de relaciones bastante completo en el que están presentes el versionado y la realización (relación semántica entre assets clasificados en niveles de abstracción diferentes), y Microsoft Repository [3] que presenta tres clases de connotaciones semánticas en las relaciones soportadas: nombre, secuencia y propagación de eliminación de los assets relacionados.

3. Relaciones en el modelo de reutilización de los mecanos

Un mecano es un elemento software reutilizable de una granularidad tal que implica que está compuesto por una serie de assets clasificados en diferentes niveles de abstracción y relacionados entre sí, formando un subsistema software.

Al hablar de relaciones entre assets se está haciendo referencia a los enlaces semánticos entre los objetos del universo de discurso, o sea el mundo del desarrollo del software, en el que se plantea la definición de los mecanos.

Cada uno de los enlaces que se mantiene entre dos assets debe pertenecer a un tipo de relación semántica que recoja las restricciones características de la familia de enlaces a la que da lugar. Esto coincide con el concepto de *power type* [13].

Para capturar de una forma adecuada la semántica de los tipos de relaciones entre assets se ha decidido tomar como base de referencia un modelo objeto que aporte un conjunto reducido de relaciones estructurales que sirvan para caracterizar los tipos de relaciones semánticas entre los assets.

En la Figura 1 se presenta un ejemplo de un mecano muy simple, en el que se pueden apreciar los tres niveles de abstracción en el que se definen las relaciones semánticas entre los assets componentes de un mecano. En el nivel más bajo de abstracción están los enlaces entre los assets (R0001 y R0002), cada una de las cuales tiene un tipo de relación semántica asociado (Implementa y Herencia, respectivamente), pero a su vez cada uno de los tipos de relación semántica está soportado por una relación estructural (Reificación y Extiende, en el ejemplo considerado).

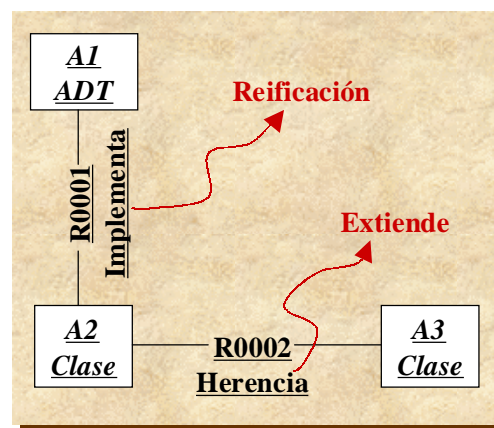


Figura 1. Diagrama de objetos que representa un mecano.

El ejemplo mostrado en la Figura 1 se utiliza en la Figura 2 para ilustrar los diferentes niveles de abstracción que se utilizan para definir los mecanos como estructuras complejas de reutilización. El nivel de modelo presente en la Figura 2 se corresponde con una parte del modelo de mecano inicialmente presentado en [10] y refinado en [11].

Cada una de las relaciones estructurales a considerar debe concentrar en su definición unas connotaciones semánticas, que serán compartidas por todos los tipos de relaciones semánticas soportados por ellas, convirtiéndose así en la base constructiva y de soporte a la generación de los mecanos.

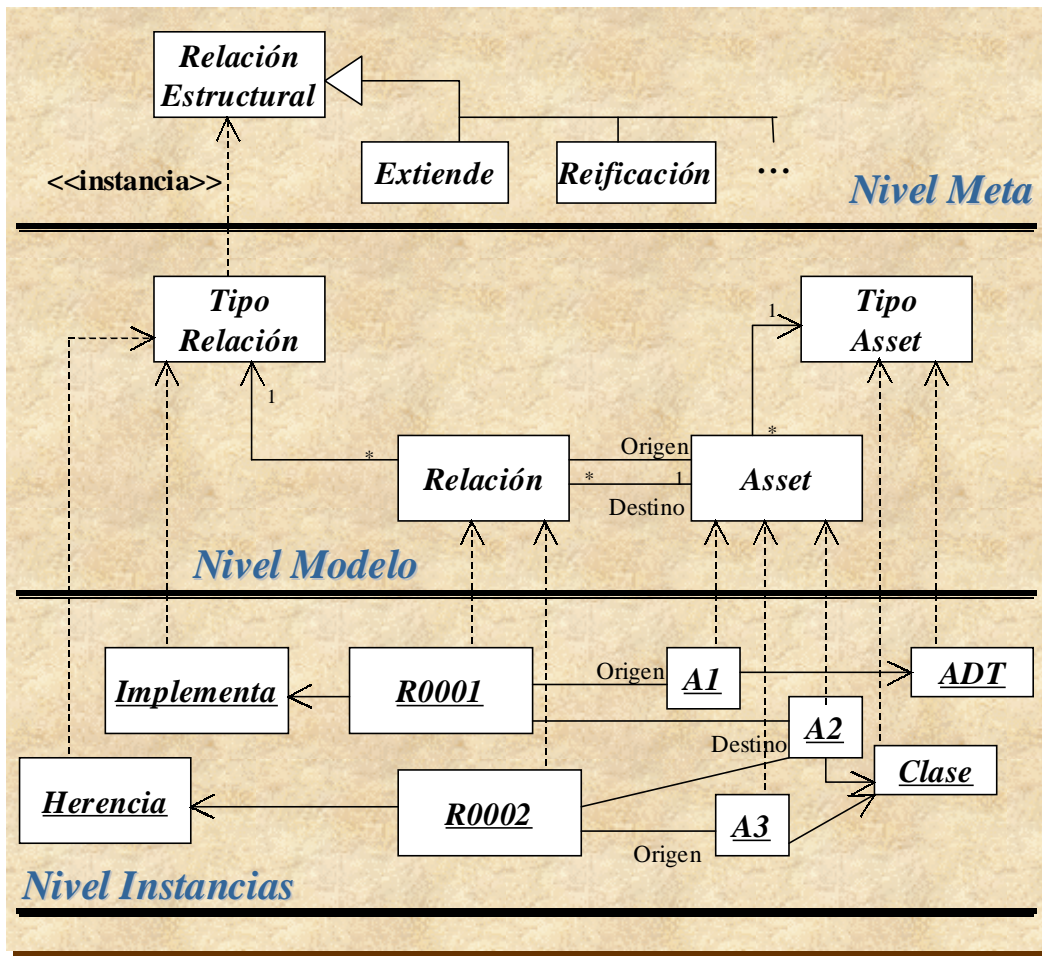


Figura 2. Niveles de abstracción en la definición de los mecanos.

3.1 Definición de las relaciones estructurales

Uno de los aspectos más relevantes de los mecanos como estructura compleja de reutilización es su definición en diferentes niveles de abstracción simultáneamente, especialmente cuando se está haciendo referencia a los mecanos que se almacenan en el repositorio como elementos software reutilizables complejos. Por tanto, al existir siempre en un mecano assets clasificados en diferentes niveles de abstracción aparece la necesidad de diferenciar entre las relaciones que se dan entre assets clasificados en un mismo nivel de abstracción, relaciones intranivel, y las relaciones que enlazan assets clasificados en diferentes niveles de abstracción, relaciones internivel. Por consiguiente, es necesario que el modelo objeto subyacente aporte las relaciones estructurales adecuadas para dar soporte a ambos tipos de relaciones.

La primera relación estructural a considerar es la que viene impuesta por la restricción de que todo mecano debe contar con al menos dos assets clasificados en niveles de abstracción diferentes y relacionados entre sí [11]. Así, se necesita una relación estructural que soporte la semántica derivada del cambio de nivel de abstracción de un asset. Esta relación es la

reificación, que expresa un refinamiento por el que se obtienen elementos software cada vez más concretos [7].

Desde el punto de vista del espacio de reutilización que se está construyendo, la reificación puede definirse como:

Definición 1 – Reificación: *Se dice que un asset Y reifica a un asset X cuando el asset Y se puede considerar como un refinamiento del asset X, y el asset X está clasificado en un nivel de mayor abstracción que el asset Y.*

Con la reificación se cubre el espectro estructural de las relaciones internivel. Pero, en lo que respecta a las relaciones intranivel se tiene una mayor diversidad semántica. En consecuencia, en el espacio de reutilización se hace necesario un conjunto más amplio de relaciones estructurales que recojan las connotaciones semánticas siguientes: “ser parte de” o relaciones de meronimia, uso, extensión y asociación.

La semántica de inclusión se hace necesaria en el espacio de reutilización para representar la relación existente entre aquellos assets de granularidad gruesa que contienen un conjunto de assets de grano más fino. Las connotaciones semánticas que existen en torno a este tipo de relaciones de inclusión son muy variadas [13]. No obstante, para los intereses de este trabajo se distinguen sólo dos tipos de relaciones estructurales de inclusión, la agregación y la composición. La agregación presenta una contención débil, donde las partes componentes tienen un ciclo de vida independiente del asset agregado, pudiendo pertenecer a varios assets agregados a la vez. Mientras que la composición implica una pertenencia más fuerte de las partes al asset compuesto, no teniendo sentido la reutilización de las partes por separado, ni su pertenencia a otro asset compuesto.

Así pues, y desde el punto de vista del espacio de reutilización, la agregación y la composición se pueden definir como:

Definición 2 – Agregación: *Se dice que un asset Y está relacionado con un asset X mediante una relación de agregación cuando el asset Y forma parte de la estructura de X, y además puede reutilizarse con independencia de la reutilización de X, estando X e Y clasificados en el mismo nivel de abstracción.*

Definición 3 – Composición: *Se dice que un asset Y está relacionado con un asset X mediante una relación de composición cuando el asset Y forma parte exclusivamente de la estructura de X, no teniendo sentido la reutilización del asset Y sin la reutilización del asset X, estando X e Y clasificados en el mismo nivel de abstracción.*

La semántica de uso representa las relaciones tipo cliente/servidor donde un asset especifica un comportamiento (servicios) que otro asset cliente utiliza sin que el primero tenga que tener un conocimiento expreso de ello. Este tipo de comportamiento o de relaciones entre asset se representa por la relación estructural “usa a”, que dentro del espacio de reutilización puede definirse como:

Definición 4 – Usa a: *Un asset Y usa un asset X cuando el asset Y depende del comportamiento especificado por el asset X para su definición o para completar su comportamiento, estando X e Y clasificados en el mismo nivel de abstracción.*

La semántica de extensión caracteriza todas aquellas relaciones en las que un asset se define en función de la definición ofrecida por otro asset. Este tipo de relaciones entre assets se representa por la relación estructural extensión, que dentro del espacio de reutilización se define como:

Definición 5 – Extensión: *Un asset Y extiende otro asset X cuando Y está definido en función de la definición de X, estando X e Y clasificados en el mismo nivel de abstracción.*

Las asociaciones se necesitan en el espacio de reutilización para expresar enlaces entre assets clasificados en el mismo nivel de abstracción, pero de manera que expresen los contenidos semánticos no soportados por los tipos de relaciones estructurales anteriores. La relación estructural asociación se define en el espacio de reutilización como:

Definición 6 – Asociación: *Un asset X está asociado con un asset Y cuando existe un enlace semántico bidireccional entre ambos assets, estando X e Y clasificados en el mismo nivel de abstracción.*

3.2 Restricciones de las relaciones

Una vez definidas las relaciones estructurales necesarias en el espacio de reutilización de definición de los mecanos, es necesario presentar una serie de restricciones que las caractericen. Estas propiedades son: *el orden de las relaciones, la dirección de las relaciones y el carácter de las relaciones frente al proceso de reutilización.*

3.2.1 Orden de las relaciones

Para el modelado de las relaciones entre los assets constituyentes de un mecano se ha optado por la utilización de relaciones binarias, como puede deducirse de las definiciones enunciadas en el apartado 3.1. Aparte de la naturalidad y sencillez que aportan a los modelos las relaciones binarias, la decisión tomada se ve refrendada por los siguientes hechos:

- Lo que interesa expresar con las relaciones entre assets es la trazabilidad de unos a otros.
- El uso de relaciones binarias es una práctica común en el modelado de sistemas, teniendo destacados ejemplos en el estándar BRM (Binary Relations Model) [1] o en el estándar ODMG 2.0 [5]. Además, este mismo criterio ha sido adoptado ya en reutilización, siendo ejemplos representativos de ello las relaciones entre componentes definidas por el RSRG [8], o las relaciones entre los objetos soportadas por el Repositorio Microsoft [3].

3.2.2 Dirección de las relaciones

Las relaciones estructurales identificadas en el espacio de reutilización se definen entre un origen y un destino, esto implica que aunque los enlaces semánticos entre los assets que forman los mecanos puedan ser recorridos en ambas direcciones, la semántica expresada por la relación es sensitiva a la *polaridad* de la relación indicada por el origen y el destino.

Por tanto, en todas las relaciones estructurales consideradas se tiene una polaridad semántica que se recoge en la Tabla 1, a excepción de la asociación, bidireccional por definición [4].

Relación	Polaridad
----------	-----------

Si se aplican las definiciones de enlace fuerte y débil para determinar el carácter de los enlaces en la relación de reificación se tiene que ésta da lugar a dos enlaces semánticos débiles, pues un asset clasificado en un determinado nivel de abstracción puede reutilizarse independientemente de los assets con los que se relaciona en otros niveles de abstracción. Sin embargo, si se presenta el problema desde el punto de vista de la generación de mecanos, interesa cualificar al menos un enlace como fuerte, el dirigido desde el asset de mayor nivel de abstracción al asset de menor nivel de abstracción.

Con la introducción de un enlace fuerte en la reificación se está asegurando la generación de mecanos en tiempo de reutilización como respuesta a las peticiones de los desarrolladores con reutilización, dado que bajo esta perspectiva generativa, la extracción del repositorio de un asset clasificado en un nivel de abstracción superior al de implementación siempre se verá acompañada de la extracción de los assets clasificados en niveles de abstracción menores relacionados con él.

Esta aproximación establece un marco de trabajo general para el enfoque generativo soportado por el modelo de reutilización. No obstante, esta forma de actuación puede verse modificada por las restricciones introducidas en las peticiones de los usuarios.

La agregación representa una semántica de contención no excluyente, por tanto reutilizar el asset de grano grueso implica reutilizar los assets que lo forman, pero la reutilización de cualquiera de sus partes no conduce a la reutilización del asset agregado, dándose un enlace fuerte del asset de grano grueso a cada uno de los assets de grano fino y un enlace débil de cada asset de grano fino al asset de grano grueso.

La composición implica enlaces fuertes en las dos direcciones debido a la dependencia por existencia de las partes con respecto al todo, como se indica en la definición.

La relación de uso presenta un enlace fuerte del cliente al servidor y un enlace débil del servidor al cliente, debido a que el cliente forzosamente tiene que conocer la existencia del servidor, pero esto no sucede en el caso opuesto.

La relación extiende presenta una dependencia del asset que extiende la definición con respecto al asset origen de la definición, por tanto la reutilización del asset derivado necesitará de la presencia del asset base que le proporciona la definición pero no al contrario, estableciéndose un enlace fuerte del derivado al base y un enlace débil del asset base al derivado.

La asociación presenta siempre enlaces débiles en las dos direcciones de navegación, lo cual permite introducir una serie de tipos de relaciones semánticas muy importantes en el espacio de reutilización que, aunque no tienen una implicación directa en la generación de mecanos, son imprescindibles en la definición de elementos software reutilizables complejos, como por ejemplo el caso de la relación de versionado.

R. Estruct.	Configuración		Tipo R. Semántica			Explicación
	Orig. ⇒ Dest.	Dest. ⇒ Orig.	Nombre	T.Ass.O	T.Ass.D	
<i>Reificación</i>	Fuerte	Débil	Implementa	ADT	Clase C++	Reutilizar el ADT implica extraer la clase C++ (en el enfoque generativo de mecanos), pero no viceversa
<i>Agregación</i>	Fuerte	Débil	Incluye	Modelo Ambiental	Diagrama de Contexto	Reutilizar el modelo de ambiente implica reutilizar el Diagrama de Contexto pero no viceversa
<i>Composición</i>	Fuerte	Fuerte	SeComponeDe	Biblioteca de Funciones	Función	Reutilizar una biblioteca de funciones supone reutilizar todas las funciones contenidas en ella, y reutilizar una función supone reutilizar la biblioteca que la contiene
<i>Uso</i>	Fuerte	Débil	SeDefineEn	DFD	Dic. De Datos	Reutilizar un DFD implica reutilizar el DD donde se definen sus flujos y almacenes, pero no viceversa
<i>Extiende</i>	Fuerte	Débil	Realiza	Clase Concreta	Clase Abstracta	Reutilizar la clase concreta requiere reutilizar la clase abstracta de la que se deriva, pero no al contrario
<i>Asociación</i>	Débil	Débil	Versión	Clase	Clase	Reutilizar una versión de una clase no implica reutilizar las versiones derivadas de ella ni sus ancestros

Tabla 2. Carácter de las relaciones estructurales en el proceso de generación de mecanos.

La Tabla 2 recoge para cada una de las relaciones estructurales del espacio de reutilización en el que se definen los mecanos, su configuración de enlaces con respecto al proceso de generación de mecanos, así como un ejemplo de un tipo de relación semántica que de ella se deriva, indicando los tipos de assets origen y destino de dicho tipo de relación semántica.

4. Conclusiones

En este trabajo se ha establecido el marco estructural para la definición de un espacio de reutilización en el que se definen unos elementos software reutilizables denominados mecanos. Este espacio de reutilización se basa en dos conceptos fundamentales los assets y sus interrelaciones.

En lo referente a las relaciones entre assets, se ha establecido y justificado un conjunto reducido de relaciones estructurales, que sirven para definir los tipos de relaciones semánticas que se pueden dar entre los tipos de assets, y por consiguiente marcan las connotaciones semánticas de los enlaces entre los assets componentes de un mecano.

Otro aspecto importante de este trabajo es la definición de un modelo de reutilización propio, basado en un enfoque mixto de composición y generación de mecanos, que cubre todo el espectro de actividades de los procesos de desarrollo para y con reutilización.

Las relaciones entre assets son fundamentales en la creación de mecanos, ya sea por composición o por generación, convirtiéndose en el elemento directriz del proceso de generación gracias a la cualificación de su carácter de fortaleza frente al proceso de reutilización. Dentro de este carácter, se ha diferenciado un matiz de obligatoriedad que mantiene el sentido de la reutilización y que se recoge en las relaciones intranivel (a excepción de la asociación), y un matiz de apoyo a la generación de mecanos reflejado en las relaciones internivel, esto es, en la reificación.

5. Referencias

- [1] **Abrial, J. R.** “*Data Semantics in Database Management Systems*”. J.W. Klimbie and K. L. Koffeman, eds., North Holland, 1974.
- [2] **Ader, M., Nierstrasz, O., McMahon, S., Muller, G. and Präfrock, A-K.** “*The ITHACA Technology: A Landscape for Object-Oriented Application Development*”. In the proceedings of ESPRIT’90 Conference. Kluwer Academic Publisher. November, 1990.
- [3] **Bernstein P. A., Harry, B., Sanders, P., Shutt, D. and Zander J.** “*The Microsoft Repository*”. VLDB’97 Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece. Morgan Kaufmann, 1997.
- [4] **Booch, G.** “*Object-Oriented Analysis and Design with Applications*”. 2nd Ed. Benjamin Cummings, 1994.
- [5] **Cattell, R. and Barry, D.** “*The Object Database Standard: ODMG 2.0*”. Morgan Kaufmann. 1997.
- [6] **Constantopoulos, P., Jarke, M., Mylopoulos, J. and Vassiliou, Y.** “*The Software Information Base: A Server for Reuse*”. ITHACA.FORTH.92.E2#1. 1992.
- [7] **Denker, G.** “*Reification - Changing Viewpoint but Preserving Truth*”. In Recent Trends in Data Types Specification, Proc. 11th Workshop on Specification of Abstract Data Types joint with the 8th General COMPASS Meeting. Oslo, Norway, September 1995. Selected papers, pages 182-199. Springer, LNCS 1130, 1996.
- [8] **Edwards, S. H., Gibson, D. S., Weide, B. W. and Zhupanov, S.** “*Software Component Relationships*”. In the proceedings of the WISR 8, 1997.
- [9] **Eichmann, D., Price, M., Terry, R. H. and Welton, L. L.** “*ELSA and MORE: A Library and an Environment for the Web*”. <http://rbse.mountain.net/MOREplus/ELSAandMORE>. 1995.
- [10] **García, F. J., Marqués, J. M. y Maudes, J. M.** “*Mecano: Una Propuesta de Componente Software Reutilizable*”. En las actas de las II Jornadas de Ingeniería del Software (Donostia-San Sebastián, España, 3-5 septiembre de 1997): 232-244. 1997.
- [11] **García, F. J., Marqués, J. M., Laguna, M. Á. y Maudes, J. M.** “*Estructuras Complejas de Reutilización: Definición de Mecano Estático*”. En las actas de las II Jornadas de Trabajo MENHIR. Editor José A. Carsí (Valencia, 19-20 de Febrero de 1998): 135-141. 1998.
- [12] **Karlsson, E.** “*Software Reuse. A Holistic Approach*”. John Wiley & Sons Ltd. 1995.
- [13] **Martin, J. and Odell, J. J.** “*Object-Oriented Methods: A Foundation*”. Prentice-Hall, 1995.
- [14] **NHSE.** “*The Internal Data Format of RIB*”. NHSE. October 24, 1997.
- [15] **SER Cons.** “*Solutions for Software Evolution and Reuse*”. SER Esprit Project 9809. Jan., 1996.
- [16] **Tracz, W.** “*Where Does Reuse Start?*”. In the proceedings of the Realities of Reuse Workshop, Syracuse University CASE Center. January, 1990.