Revista de Procesos y Métricas de las Tecnologías de la Información (RPM) ISSN: 1698-2029
VOL. 1, Nº 3, Diciembre 2004, 29-38
Asociación Española de Sistemas Informáticos (AEMES)

UN PROCEDIMIENTO DE MEDICIÓN DE TAMAÑO FUNCIONAL: DISEÑO Y APLICACIÓN

Nelly Condori-Fernandez[1], Silvia Abrahão[1],Oscar Pastor[1],Sofia Martí[2]
Departamento de Sistemas Informáticos y Computación[1]
Universidad Politécnica de Valencia, Camino de Vera s/n, 46022, Valencia.
E-Mail :{nelly,sabrahao,opastor @}dsic.upv.es

CARE Technologies, S. A, Ctra. Las Marinas [2]
km. 2 03700 Denia, Spain +34 96 642 6970
E-Mail : smarti@care-t.com

Abstract: In this paper, we present the process followed for the design of a functional size measurement procedure for object-oriented systems from requirements specifications expressed in OO-Method, an automatic software production method. This measurement procedure was designed from the generic metamodel of the COSMIC-FFP functional size measurement method. In addition, a case study is presented with the purpose of illustrating the application of the representation, de-duplication and measurement rules defined as part of the proposed measurement procedure.

Resumen: En este trabajo se presenta el proceso seguido para el diseño de un procedimiento de medición de tamaño funcional para sistemas orientado a objetos a partir de especificaciones de requisitos expresados en OO-Method, un método de producción automático de software. Este procedimiento de medición fue diseñado siguiendo el metamodelo genérico del método de medición de tamaño funcional COSMIC-FFP (COSMIC-Full Function Point). Además, un caso de estudio es presentado con el propósito de ilustrar la aplicación de las reglas de representación, de duplicidad y de medición definidas como parte del procedimiento de medición propuesto.

Palabras Clave: Procedimiento de medición, especificación de requisitos, tamaño Funcional, OO-Method y COSMIC-FFP.

1. INTRODUCCIÓN

La medición del tamaño del software es actualmente un medio esencial para realizar las estimaciones oportunas de diversos indicadores necesarios para el desarrollo de productos software. De este modo, la precisión de los instrumentos de medición juega un rol importante porque se constituyen en herramientas de soporte para los directores de proyectos de desarrollo de software. Un instrumento de medición no es más que la automatización de un procedimiento de medición [1] previamente definido. Un diseño sistemático de dicho procedimiento facilitará la automatización y la obtención de instrumentos de medición más fiables.

[1] "Conjunto de operaciones descritas de manera específica y usadas en la ejecución de una medición particular de acuerdo a un método de medición dado" [21].

En este artículo se presenta los pasos del diseño y la aplicación de un procedimiento de medición de tamaño funcional para sistemas orientados a objeto. En particular, dicho procedimiento fue diseñado en el contexto del Modelo de Requisitos de OO-Method [16] como resultado de la instanciación del método de medición de tamaño funcional COSMIC-FFP [10]. Este método ha sido aprobado como un estándar por la ISO/IEC 19761:2003 [22] y es considerado como un método de segunda generación. COSMIC-FFP es aplicable en la medición de arquitecturas de software multicapa y debido a su carácter genérico es también aplicable a diferentes dominios de software. Además, existe un fuerte impulso por parte de COSMIC y del grupo ISBSG (Internacional Software Benchmarking Standards Group) por recoger datos de proyectos medidos con COSMIC-FFP [23] y fortalecer de esta manera su aplicabilidad en la industria del software.

Actualmente, existen varias propuestas para instanciar la genericidad de COSMIC-FFP [10], tales como Bevo [8] y Jenner [9] quienes realizan una correspondencia directa para los conceptos de UML [3], tales como los

diagramas de casos de uso, de clases y de secuencia. Sin embargo, la propuesta de Jenner no identifica con claridad los movimientos de datos de tipo escritura y lectura en los mensajes de los diagramas de secuencia.

Poels [24] ha propuesto un procedimiento de medición que describe cómo los conceptos del metamodelo de COSMIC-FFP pueden ser aplicados en el contexto del método de desarrollo MERODE (Model-driven Existence-dependency Related Object-oriented Development). En este trabajo la estimación de tamaño se obtiene a partir de modelos conceptuales multicapa. De igual manera, la propuesta de Diab [25] describe una correspondencia entre los conceptos COSMIC-FFP y las primitivas conceptuales de un método de desarrollo de sistemas de tiempo real denominado ROOM (Real-Time Object-Oriented Modelling). Ambas propuestas estiman el tamaño funcional en una etapa más tardía a la etapa de especificación de requisitos.

Además, existen también otras propuestas cuyo propósito es estimar el tamaño funcional en etapas tempranas del ciclo de desarrollo de software (Análisis de Requisitos) de acuerdo a otros métodos de medición. Entre ellas se encuentran la propuesta de Fetcke y otros [1] quienes relacionan los conceptos de IFPUG FPA (Function Point Analysis) [2] para el método de desarrollo OOSE (Object Oriented Software Engineering) [3]. Un trabajo similar ha sido propuesto por Tavares [4] quien define un conjunto de reglas para relacionar las primitivas de los diagramas de casos de uso y diagramas de clases UML [20] con los conceptos del método IFPUG FPA. Bertolami [5] considera el método de medición de tamaño funcional Mark II FPA [7] para medir en la etapa de elicitación de requisitos a partir de escenarios construidos con LEL [6], un Lenguaje Extendido de Lexicones.

Todas estas propuestas carecen de un proceso que guíe el diseño de los respectivos procedimientos de medición. Para cubrir dicha carencia, nosotros utilizamos el Modelo de Proceso de Medición propuesto por Jacquet y Abran [11], el cual comprende de cuatro etapas: *Diseño del procedimiento de medición, Aplicación de dicho procedimiento, Análisis de los resultados de medición y Exploración de los resultados*. Este trabajo comprende la descripción de las dos primeras etapas.

Por lo tanto, el presente artículo está organizado de la siguiente manera: La sección 2 describe el Modelo de Requisitos de OO-Method. La sección 3 presenta los pasos seguidos para el diseño del procedimiento de medición de tamaño funcional. La sección 4 ilustra la aplicación del procedimiento de medición en un caso de estudio. Finalmente, en la sección 5 presentamos las conclusiones y trabajos futuros.

2. MODELO DE REQUISITOS DE OO-METHOD

OO-Method es un método orientado a objetos de producción automática de software [18]. Tal como se observa en la Figura 1, este método aborda la tarea de construcción del software utilizando tres modelos a lo largo del espacio del problema y del espacio de la solución: Modelo de Requisitos, Modelo Conceptual y Modelo de Ejecución.

El Modelo de Requisitos captura los requisitos funcionales del software y por medio de un proceso de análisis se procede con la generación semi-automática de los esquemas conceptuales y a partir de dichos esquemas se procede con la generación automática del software a través del Modelo de Ejecución, previa compilación de la especificación formal basada en el lenguaje OASIS [19].

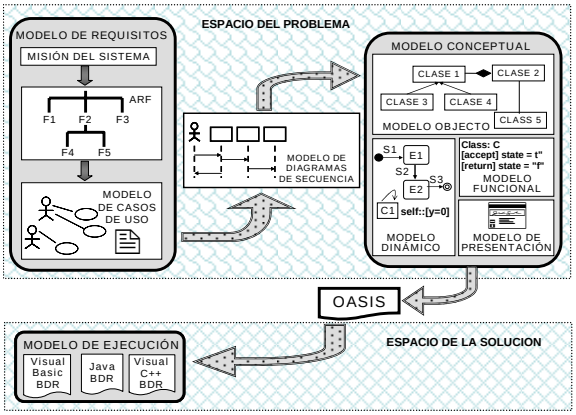

Figura 1. Proceso de desarrollo de OO-Method

A continuación se presenta el Modelo de Requisitos por ser relevante para el presente trabajo.

2.1 Modelo de Requisitos

El Modelo de Requisitos [16] comprende de tres técnicas que permiten la captura de la funcionalidad del sistema que se desea construir. Estas son: la Misión del Sistema, el Árbol de Refinamiento de Funciones y el Modelo de Casos de Uso.

La *Misión del Sistema* describe el propósito del sistema. A través de la definición de su misión es posible determinar qué hará y qué no hará el sistema.

El *Árbol de Refinamiento de Funciones* representa la descomposición jerárquica de las funciones de un sistema. Este árbol es una organización de interacciones externas y es una técnica muy útil para

la construcción del Modelo de Casos de Uso con un nivel de abstracción adecuado.

El *Modelo de Casos de Uso* permite modelar los requisitos funcionales del sistema desde la perspectiva del usuario. Si los casos de uso tienen una correspondencia directa con los nodos hoja del Árbol de Refinamiento de Funciones (función elemental), se consideran como primarios, estos representan las funciones más importantes del sistema. Por lo contrario, los casos de uso secundarios surgen con la finalidad de descomponer la complejidad de los casos de uso primarios mediante relaciones de EXTEND e INCLUDE.

2.2 Modelo de Diagrama de Secuencia

Una vez obtenido el Modelo de Requisitos, los casos de uso son refinados y expresados en términos de responsabilidades. Estas responsabilidades son descritas como clases que cooperan entre sí para la obtención de las funcionalidades especificadas en los casos de uso. Para expresar las responsabilidades a este nivel de detalle se utiliza los *Diagramas de Secuencia*, los cuales permiten describir patrones de interacción mediante un conjunto de mensajes. Básicamente se distingue cuatro tipos de mensajes:

Mensajes de Señal: Su estereotipo es «signal» y representan una interacción entre el actor y el sistema. La única propiedad que discrimina este tipo de mensajes es la dirección del mensaje, la cual puede ser de dos tipos de valores: *input* y *output*.

Mensajes de Servicio: Son estereotipados con la etiqueta «service» y representan la modificación de estado de la clase receptora del mensaje cuando ocurre una interacción. El tipo de cambio puede ser de tres tipos: *new*, *destroy* y *update*.

Mensajes de Consulta: Son estereotipados con la etiqueta «query» y representan consultas sobre el estado de un objeto cuyo resultado es un valor calculado o consultas sobre una población de la clase receptora cuyo resultado es un listado de objetos.

Mensajes de Conexión: Son estereotipados con la etiqueta «connect» y se utilizan para establecer una relación estructural entre los objetos participantes en la interacción.

Estos mensajes adicionalmente pueden ser etiquetados con alguna condición que si es satisfecha se permite la interacción. La sintaxis para esta condición es:

[Bolean-expresión] message-name.

Cuando un conjunto de mensajes es implicado por alguna condición o iteración, este conjunto de mensajes es representado gráficamente por una caja denominada *bloque*.

Finalmente, tal como se observa en la Figura 1, estos diagramas de secuencia actúan como un puente entre el Modelo de Requisitos y el Modelo Conceptual.

3. DISEÑO DEL PROCEDIMIENTO DE MEDICIÓN

En esta sección, se presenta los pasos del Modelo de Proceso de Medición [11] seguidos para el diseño de un procedimiento de medición. Estos pasos son: la definición de objetivos, la caracterización del concepto a ser medido, la selección del metamodelo del dominio y la definición de reglas de asignación numérica.

3.1 Definición de Objetivos

Este primer paso nos permite saber qué deseamos medir (objeto, atributo), bajo qué punto de vista será definido el procedimiento de medición y cuáles son las aplicaciones previstas de dicho procedimiento a definir. Para responder a estas interrogantes se utilizó una plantilla de medición de software orientado a metas denominado GQM [12]. Por consiguiente, el objetivo a alcanzar es:

Definir *un Procedimiento de Medición de Tamaño Funcional* ***para el propósito de*** *estimar el tamaño funcional de los sistemas OO* ***con respecto a la*** *especificación de requisitos de software* ***desde el punto de vista del*** *Investigador* ***en el contexto de*** *OO-Method.*

3.2 Caracterización del concepto a ser medido

Este paso tiene como fin definir el atributo a ser medido. En nuestro procedimiento a diseñar, se tiene como atributo al *tamaño funcional,* el cual es definido por la ISO/IEC 14143-1 [14] como el tamaño del software derivado por la cuantificación de los requisitos funcionales de usuario. Sin embargo, es necesario precisar esta definición de acuerdo al metamodelo seleccionado. Es decir, que teniendo en cuenta el método de medición COSMIC-FFP [10], el tamaño funcional es *el número de movimientos de datos identificados*. A continuación, describimos el metamodelo seleccionado.

3.3 Selección del Metamodelo

La finalidad de este paso es identificar los elementos que son relevantes para la medición y construcción de un modelo de software de acuerdo a un método de medición seleccionado. La Figura 2, representa el

metamodelo de COSMIC-FFP, el cual ha sido diseñado a partir del respectivo manual de medición [10]. A continuación describimos cada uno de los conceptos relevantes y las relaciones presentadas en el metamodelo de COSMIC-FFP, (de abajo hacia arriba).

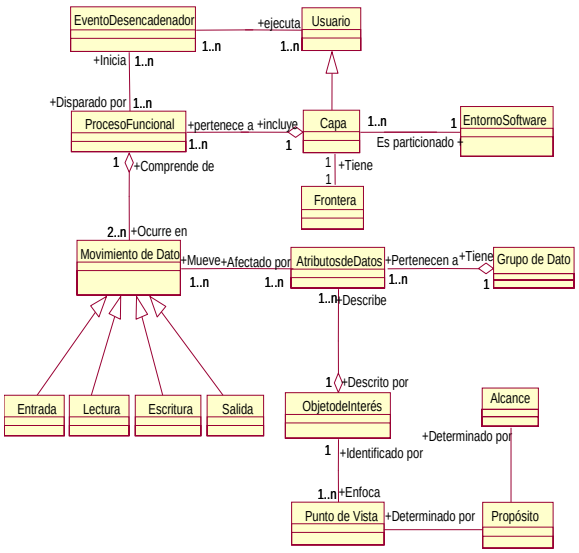

Figura 2. Metamodelo del Método de Medición COSMIC-FFP

El *propósito de medición* define el porqué y para qué de la medición, permite la determinación del alcance y el punto de vista de la medición. El *alcance de la medición* representa toda la funcionalidad a ser incluida en un proceso de medición de tamaño funcional. Y el *punto de vista de medición* precisa el nivel de detalle con que puede ser enfocado un objeto. Este *objeto de interés* a ser medido puede ser alguna cosa física descrita por un conjunto de atributos de datos. Un *atributo de dato* es la pieza más pequeña de información que pertenece a un grupo de datos y son afectados por uno o varios movimientos de datos. Un *movimiento de datos* ocurre en un proceso funcional y puede ser de cuatro tipos: entrada, lectura, escritura y salida. Todo *proceso funcional* es disparado por uno o varios eventos desencadenadores ejecutados de manera directa o indirecta por un usuario. Un *usuario* es definido como una persona o cosa que se comunica o interactúa con el software. En un entorno de software multicapa, cada capa usa los servicios funcionales proveídos por otras capas subordinadas. El concepto de *capa* es representado como una especialización del concepto usuario y es el resultado de la partición funcional del *entorno operativo de software* en función a un mismo nivel de abstracción.

3.3.1 Definición de reglas de representación y de duplicidad

Para identificar todos estos conceptos en el contexto de OO-Method, se ha definido un conjunto reglas con el fin de establecer una correspondencia entre las primitivas del Modelo de Requisitos de OO-Method [16] y los conceptos del Metamodelo de COSMIC-FFP [10]. La definición de este conjunto de reglas ha sido explicada y discutida con mayor detalle en [13]. En la Tabla 1 se enumera las reglas respectivas para cada una de las primitivas relevantes del Modelo de Requisitos de OO-Method.

Tabla 1. Reglas de representación: COSMIC-FFP y OO-Method

Conceptos COSMIC-FFP		Primitivas OO-Method
Usuarios		**Regla 1:** Actores
Frontera		**Regla 2**: Diagrama de casos de uso
Procesos Funcionales		**Regla 3:** Casos de Uso Primarios **Regla 4**: Casos de uso Secundarios
Grupos de datos		**Regla 5:** Actores **Regla 6:** Clases
Atributos de datos		**Regla 7**: Atributos de Clases
Movimientos de Datos	Entrada	**Regla 8:** Mensaje *Signal* con valor Input
	Lectura	**Regla 9**: Mensaje *Query* **Regla 10:** Condición de Mensaje **Regla 11**: Precondición **Regla 12:** Condición de relación EXTEND
	Escritura	**Regla 13:** Restricción de integridad **Regla 14:** Mensaje *Service* con propiedad *New*, *Destroy* o *Update* **Regla 15:** Mensaje *Connect*
	Salida	**Regla 16:** Mensaje *Signal* con valor *Output*

Los conceptos *propósito*, *punto de vista* y *alcance de la medición* permiten determinar el contexto de la medición. Estos conceptos son necesarios para definir el porque de la medición y delimitar el artefacto a ser medido. Sin embargo, no tienen una correspondencia directa con las primitivas del Modelo de Requisitos de OO-Method.

Por otro lado, con la finalidad de mejorar la reproducibilidad de la medición se ha complementado este paso con un conjunto de reglas para evitar la duplicidad en la medición de movimientos de datos.

En la Tabla 2 se presentan dichas reglas:

Tabla 2. Reglas de duplicidad

Primitivas	Reglas de duplicidad
Mensajes de interacción	**Regla 17:** Los n-mensajes del mismo tipo relacionados al mismo objeto de la clase receptora es considerado como sólo un movimiento de dato.
Mensajes de error y confirmación	**Regla 18:** Los mensaje de error y confirmación especificados en un proceso funcional son identificados como sólo un movimiento de dato tipo salida.
Bloque de un diagrama de secuencia	**Regla 19:** El conjunto de movimientos de datos identificados en un bloque deben ser considerados una sola vez. **Regla 20**: Lo mensajes INCLUDE que pertenecen a un bloque deben ser considerados una sola vez.

3.4 Definición de reglas de asignación numérica

La finalidad de este último paso es producir un valor cuantitativo que represente el tamaño funcional de las especificaciones de requisitos de OO-Method. Para esto, se aplica la *función de medición* que consiste en asignar un tamaño numérico de 1 Cfsu (*COSMIC Functional Size Unit*) a cada instancia de movimiento de dato identificado (entrada, lectura, escritura y salida).

Además, con la finalidad de obtener el tamaño funcional de cada proceso funcional y de cada capa identificada se ha definido un conjunto de reglas correspondientes con las respectivas funciones de agregación, las mismas que fueron explicadas con mayor detalle en [13]. La Tabla 3 resume dichas reglas:

Tabla 3. Reglas de medición

Funciones COSMIC-FFP	Reglas de Medición
Función de Agregación a nivel de proceso funcional	**Regla 21:** El tamaño funcional de un caso de uso es igual a la suma de todos los movimientos de datos identificados menos uno.
	Regla 22: El tamaño funcional de un caso de uso base **extendido** por otro conjunto de casos de uso secundarios es igual a la suma de los movimientos de datos identificados en cada caso de uso secundario más los movimientos de datos del caso de uso base.
	Regla 23: El tamaño funcional de un caso de uso base que **incluye** otros casos de uso secundarios es igual a la suma de los movimientos de datos identificados en cada caso de uso incluido más los movimientos de datos del caso de uso base.
	Regla 24: El tamaño funcional de un caso de uso con **secciones alternativas** es igual al a la suma de movimientos de datos de la sección del curso básico y de la sección alternativa.
Función de Agregación a nivel de capa	**Regla 25:** El tamaño funcional de una capa de software es igual a la suma de los tamaños funcionales de todos los casos de uso primarios identificados.

4. APLICACIÓN DEL PROCEDIMIENTO DE MEDICIÓN

Con la finalidad de ilustrar la aplicación del procedimiento de medición siguiendo los pasos del modelo de proceso de Jacquet y Abran [11] se utiliza un sistema para la Gestión de Servicios de Mantenimiento (GSM) de un hospital. No obstante, este procedimiento de medición ha sido aplicado a otros casos de estudio tales como un sistema para el alquiler de vehículos [13]y un sistema para el seguimiento de torneos de golf. A continuación, presentamos los tres pasos seguidos en esta etapa: la documentación del software, la construcción del modelo del software y la aplicación de las reglas de asignación numérica.

4.1 Documentación del software

En este paso, presentamos la especificación semi-formal de los requisitos funcionales de usuario para el sistema GSM usando las siguientes técnicas del Modelo de Requisitos de OO-Method: misión del sistema, árbol de refinamiento de funciones, modelo de casos de uso y diagrama de secuencias.

La *misión del sistema* del servicio de mantenimiento es “gestionar las averías que se producen en un hospital así como otras actividades que se derivan, tales como, el mantenimiento de las máquinas del hospital, la gestión del personal del servicio, la gestión de los pedidos de las piezas necesarias para la resolución de las averías y el control de los gastos implicados”.

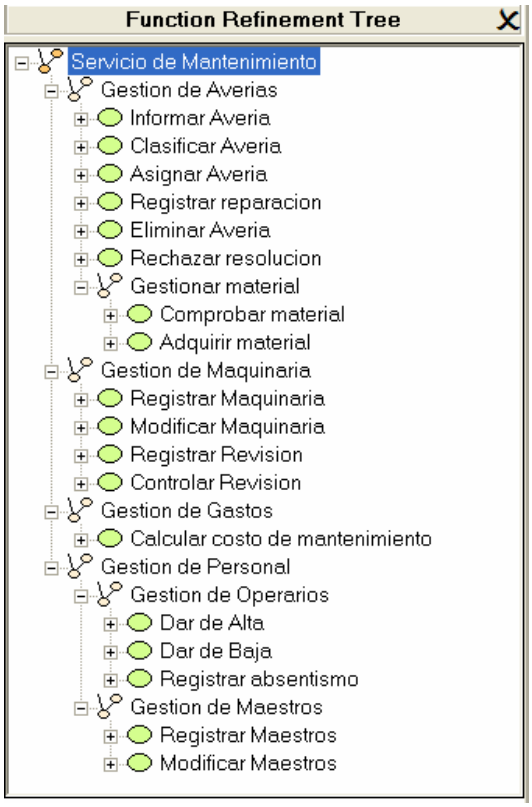

Figura 3. Árbol de Refinamiento de Funciones del sistema GSM

El *árbol de refinamiento de funciones* es construido a partir de la definición de la misión del sistema. La Figura 3 muestra los grupos funcionales y las funciones elementales del sistema GSM.

Los *casos de uso* son derivados a partir de las funciones elementales. Los *actores* identificados en el sistema son: maestro, jefe de servicio de mantenimiento, operario y personal afectado. Las relaciones que existen entre los actores y los casos de uso del grupo funcional *Gestión de Averías* son representadas en el *diagrama de casos de uso*, tal como se muestra en la Figura 4. Donde el caso de uso "eliminar avería" no presenta una relación directa con el actor maestro, y por tal motivo este caso de uso es considerado como secundario.

Finalmente, cada caso de uso es expresado en uno o varios *diagramas de secuencia*. Estos diagramas son construidos a partir de un proceso de análisis de requisitos y se constituyen en la pieza principal para la medición del sistema a desarrollar.

4.2 Construcción del modelo del software

Este paso tiene como fin identificar los elementos relevantes del modelo de requisitos de OO-Method para la construcción del modelo del software a medir de acuerdo al metamodelo definido y a la aplicación de las reglas de representación presentadas en la Tabla 1.

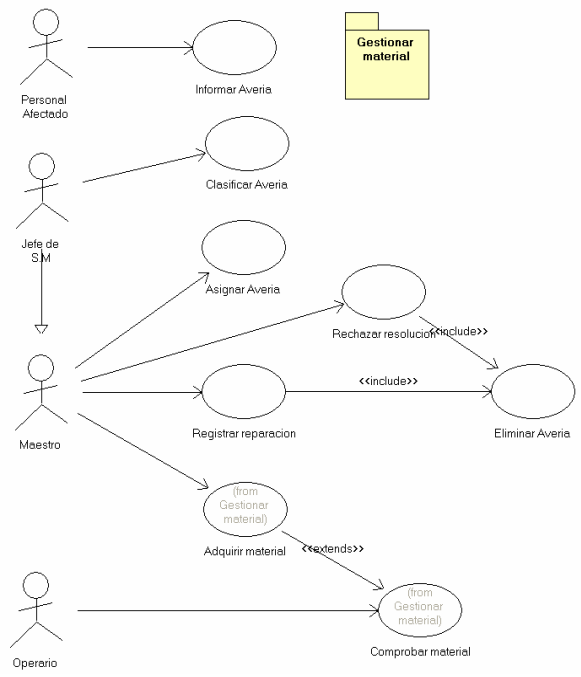

Figura 4. Diagrama de casos de uso - Gestión de Averías

A continuación ilustramos estas reglas en algunos escenarios del caso estudio descrito anteriormente.

Tal como se observa en la Figura 5, el actor maestro inicia el escenario para luego introducir el código de avería, los cuales son representados con el estereotipo «signal». Estos mensajes originan el movimiento de dato de tipo entrada (Regla 8).

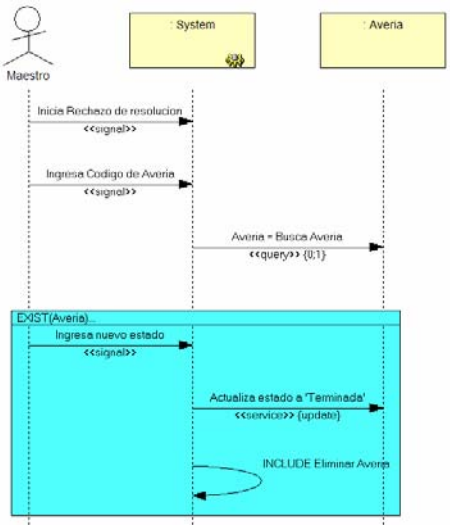

Figura 5. Rechazar resolución

Después de esta previa introducción de dato, el sistema verifica la existencia de la avería a través del mensaje con el estereotipo «query». Este mensaje es identificado como un movimiento de dato tipo lectura (Regla 9). Si existe, se introduce otros datos de avería para que el sistema actualice el objeto seleccionado. El mensaje "ingresa nuevo estado" no origina ni un movimiento de dato puesto que este mensaje involucra atributos del mismo grupo de datos (Avería) del mensaje "ingresa código de avería" (Regla 17). Por otro lado, el mensaje "actualiza estado" si origina un movimiento de dato tipo escritura (Regla 14). Luego el mensaje "INCLUDE eliminar avería" implica considerar el caso de uso "Eliminar avería" para la identificación de sus respectivos movimientos de datos (ver Figura 6).

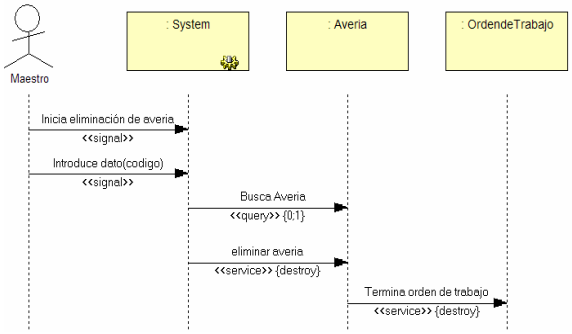

Figura 6. Eliminar avería

Además, teniendo en cuenta las REGLAS 19 y 20 los movimientos de datos y el mensaje include identificados dentro del bloque son considerados sólo una vez.

4.3 Aplicación de las reglas de asignación numérica

Finalmente, cada movimiento de dato identificado es cuantificado mediante la asignación de un valor numérico de 1 Cfsu (*función de medición*). Para obtener el tamaño funcional del caso de uso "Rechazar resolución" aplicamos en primer lugar la REGLA 21.

$$Size_p(\text{Re}chazar\_resolución) = \sum Size(data\_movement) - 1$$

$Size_p$(Rechazar_resolución) = 4 – 1

$Size_p$(Rechazar_resolución) = 3 Cfsu

Se sustrae el valor de 1 movimiento de dato por el mensaje inicial de todo escenario. Este tipo de mensaje semánticamente no es un movimiento de dato por no involucrar atributos de ningún grupo de dato identificado.

De la misma forma, aplicamos la REGLA 21 para el caso de uso "Eliminar Avería":

$$Size(E\lim inar\_averia) = \sum Size(data\_movement) - 1$$

Size(Eliminar_averia) = 5 – 1

Size(Eliminar_averia) = 4 Cfsu

Por último, aplicamos la REGLA 23 por la relación de tipo INCLUDE que presenta el caso de uso "Rechazar resolución".

$$Size(\text{Re}chazar\_resolucion) = Size(E\lim inar\_averia) + Size_p(\text{Re}chazar\_resolución)$$

Size(Rechazar_resolución) = 4 + 3

Size(Rechazar_resolución) = 7 Cfsu

Por lo tanto, el tamaño funcional del caso de uso rechazar resolución es de 7 Cfsu.

De manera similar, se procedió con el resto de procesos funcionales identificados para el sistema GSM. La Figura 7 ilustra el tamaño funcional para cada proceso funcional así como el tamaño de todo el sistema, el mismo que fue obtenido mediante la aplicación de la REGLA 24.

$$Size(Layer_1) = \sum_{i=1}^{17} Size(primary\_usecase)_i$$

*Size(Layer) = **100 cfsu***

Asimismo, los procesos funcionales *registrar reparación*, *rechazar resolución*, *comprobar material y registrar absentismo* presentan relaciones con otros casos de uso. Y el caso de uso *eliminar avería* no es considerado para el tamaño total del sistema por no ser un caso de uso primario. De esta forma, el tamaño funcional del sistema GSM es de 100 cfsu.

MOVIMIENTOS DE DATOS / PROCESOS FUNCIONALES	ENTRADA	LECTURA	ESCRITURA	SALIDA	Size : R21	Size: R22,R23	Functional Size
Informa Averia	1	0	1	1	3		3
Clasificar Averia	2	1	3	1	7		7
Asignar Averia	3	4	3	2	12		12
Registrar reparación	1	0	2	0	3	4	7
Eliminar Averia	1	1	2	0	4		
Rechazar resolución	1	1	1	0	3	4	7
Comprobar material	2	2	1	2	7	5	12
Adquirir material	3	2	2	0	7		7
Registrar maquinaria	3	0	4	0	7		7
Modificar maquinaria	1	1	1	0	3		3
Registrar revision	2	1	2	0	5		5
Controlar Revision	1	1	0	2	4		4
Calcular costo	1	2	0	1	4		4
Dar de alta	2	0	2	0	4		4
Dar de baja	2	1	1	0	4		4
Registrar absentismo	2	1	2	1	6	4	10
Registrar maestros	1	0	1	0	2		2
Modificar maestros	1	0	1	0	2		2
	CAPA 1						100

Figura 7. Resumen de la medición del tamaño funcional del sistema GSM

5. CONCLUSIONES

En este trabajo, se ha definido un procedimiento de medición para COSMIC-FFP con el objetivo de estimar el tamaño funcional de sistemas orientado a objetos a partir de especificaciones de requisitos de software. Para esto, se ha seguido el Modelo de Proceso propuesto por Jacquet y Abran [11] con la finalidad de llevar a cabo un diseño más sistemático del procedimiento de medición. La contribución principal del presente trabajo es que la estimación del tamaño funcional de los sistemas orientado a objetos se hace en una etapa temprana del proceso de desarrollo OO-Method. Se ha aplicado a varios casos de estudio lo que ha permitido el ajuste de reglas de medición y de duplicidad. Actualmente, estamos diseñando un estudio experimental con el objetivo de evaluar la reproducibilidad del procedimiento de medición propuesto. Además, la herramienta RETO[2] está siendo extendida con un módulo que permitirá obtener la estimación del tamaño funcional de manera automática mediante la automatización del procedimiento de medición definido.

REFERENCIAS

[1] T. Fetcke, A. Abran, T. Neguyen, "Mapping the OO-Jacobson Approach into Function Point Analysis" in 6th International Workshop on Software Metrics, F. Lehner , Regensburg, Germany , 1997.

[2] IFPUG, "Function Point Counting Practices Manual, Release 4.1", International Function Point Users Group, Westerville, Ohio, USA 1999.

[3] J. Rumbaugh, I. Jacobson, G. Booch, The Unified Modeling Language Reference Manual, Addison-Wesley, Reading, MA, USA, 1999.

[4] H. Tavares, A. Carvalho, J. Castro, "Medicao de Pontos por Funcao a partir da Especificacao de Requisitos", WER 2002, Valencia-España.M.

[5] Bertolami, A. Oliveros, "Proceso de Medición de Funcionalidad en la Elicitación de Requerimientos", IDEAS 2004, Arequipa-Perú.

[6] J. Leite, J. Hadad, G. Doorn, J. Kaplan, "A Scenario Construction Process", Requirements Engineering Journal, 2000.

[7] UKSMA, "MKII Function Point Analysis Counting Practices Manual. Version 1.3.1", United Kingdom Software Metrics Association 1998.

[8] V. Bévo, G. Lévesque, and A. Abran, "Application de la méthode FFP à partir dùne spécification selon la notation UML : compte rendu des premiers essais d'application et questions", Proc. Nineth Int'l. Workshop Soft. Measurement, 1999, pp. 230-242.

[9] M.S. Jenner, "COSMIC-FFP and UML: Estimation of the Size of a System Specified in UML – Problems of Granularity," Proc. Fourth European Conf. Soft. Measurement and ICT Control, pp. 173-184, 2001.

[10] COSMIC-FFP Measurement Manual version 2.2, Common Software Measurement International Consortium, January 2003.

[11] J. P. Jacquet and A. Abran, "From Software Metrics to Software Measurement Methods: A Process Model", 3rd Int. Standard Symposium and Forum on Software Engineering Standards (ISESS'97), Walnut Creek, USA, 1997.

[12] V. R. Basili and H. D. Rombach, "The TAME Project: Towards Improvement-Oriented Software Environments", IEEE Transactions on Software Engineering, vol. 14, no. 6, pp. 758-773, 1988.

[13] N. Condori-Fernández, S. Abrahao, O. Pastor, Towards a Functional Size Measure for Object-Oriented Systems from Requirements Specifications Abstract Functional Size Measurement. IEEE Quality Software International Conference 2004, Alemania.

[14] ISO, "ISO/IEC 14143-1- Information Technology -Software measurement-Functional Size Measurement. Part 1: Definition of Concepts", 1998.

[15] L. Briand, S. Morasca, V. Basili, "Property-based Software Engineering Measurement", CS-TR-3368, UMIACS-TR-94-119, University of Maryland, Computer Science Technical Report, November 1994.

[16] E. Insfran "A Requirements Engineering Approach for Object-Oriented Conceptual Modeling", PhD Thesis, Valencia Polytechnic University, October 2003.

[17] N. Condori-Fernández, Trabajo de Investigación: Validación Teórica del Procedimiento de Medición de Tamaño Funcional, Departamento de Sistemas Informáticos y Computación, 2004.

[18] O. Pastor, J. Gomez, E. Insfran, V. Pelechano: The OO-Method approach for information systems modelling: from object-oriented conceptual modelling to automated programming. Information Systems 26 (2001) 507-534.

[19] O. Pastor and I. Ramos, OASIS version 2 (2.2): A Class-Definition Language to Model Information Systems. 1995, Valencia, España: Servicio de Publicaciones Universidad Politécnica de Valencia.

[2] RETO, Requirements Tool: http://reto.dsic.upv.es/reto/

[20] J. Rumbaugh, I. Jacobson, G. Booch, The Unified Modeling Language Reference Manual, Addison-Wesley, Reading, MA, USA, 1999.
[21] ISO, 1993. International Vocabulary of Basic and General Terms in Metrology", International Organization for Standardization, Switzerland.
[22] ISO, 2003. ISO/IEC 19761 Software Engineering-COSMIC-FFP-A Functional Size Measurement Method.
[23] COSMIC-FFP Benchmarking: http://www.lrgl.uqam.ca/cosmic-ffp/benchmarking/cosmic_isbsg_bmark_annou.pdf (página accedida el 25 de febrero de 2005)
[24] Poels G., 2003. Functional Size Measurement of Multi-Layer Object-Oriented Conceptual Models, Working Paper, Gent University.
[25] Diab H., Koukane F., Frappier M., St-Denis R., 2004. µcCROSE : Automated Measurement of COSMIC FFP for Rational Rose Real Time, Information and Software Technology.