

# Construcción de un modelo de predicción para el entendimiento de los diagramas de estados en UML.

Marcela Genero<sup>1</sup>, José Antonio Cruz-Lemus<sup>2</sup> y Mario Piattini<sup>1</sup>

Grupo ALARCOS

Departamento de Informática, Universidad de Castilla-La Mancha  
Paseo de la Universidad, 4, 13071, Ciudad Real (España)

<sup>1</sup>{Marcela.Genero, Mario.Piattini}@uclm.es

<sup>2</sup>jacruz@proyectos.inf-cr.uclm.es

**Resumen.** En este trabajo se presenta una aplicación de la lógica borrosa en el ámbito de la predicción en la Ingeniería del Software. Concretamente utilizamos los Prototipos Deformables Borrosos con dos objetivos: 1) para caracterizar los diagramas de estados UML de acuerdo a su facilidad de entendimiento, partiendo de su complejidad estructural expresada mediante métricas y 2) para obtener un modelo de predicción del tiempo de entendimiento de los diagramas de estados UML. El modelo de predicción obtenido se ha construido a base de datos obtenidos en un experimento. Después de validar el modelo de predicción obtenido con datos obtenidos en otro experimento similar, se puede concluir que el modelo es en cierta manera válido ya que el 75% de los valores estimados del tiempo de entendimiento tienen al menos un 70% de exactitud, aunque es prematuro considerarlo como definitivo sin haberlo validado con datos referentes a proyectos reales.

**Palabras claves:** prototipos deformables borrosos, lógica borrosa, modelo de predicción, diagramas de estados UML, métricas, entendimiento, complejidad estructural.

## 1. Introducción

La calidad del software se ha convertido en nuestros días en un factor de discriminación entre productos, contribuyendo en gran medida al éxito de los mismos y al de la compañía que los fabrica. En la Ingeniería del Software es bien sabido que las características que hacen a la calidad de los productos software orientados a objetos (OO), como la mantenibilidad, debe ser garantizada desde las etapas iniciales de su ciclo de vida [2][7][18]. Utilizando UML [14] en el desarrollo OO, se realizan diagramas que cubren tanto aspectos estáticos como dinámicos. Para evaluar la calidad y mantenibilidad de tales diagramas de manera objetiva es necesario contar con medidas cuantitativas que eviten sesgos en el proceso de evaluación.

Existen varios trabajos sobre métricas para medir la calidad de los diagramas estructurales como los diagramas de clases [3][5][8][10][11]. Sin embargo, apenas hay unas pocas referencias en la bibliografía sobre métricas para los diagramas de comportamiento. Una de las primeras aproximaciones hacia la definición de métricas

para diagramas de comportamiento aparece en [6], donde se definen y aplican métricas para diagramas desarrollados con OMT [17]. En [16] se definen medidas de complejidad para modelos conceptuales orientados a objetos dirigidos por eventos. Sin embargo dichas propuestas de métricas no han pasado de la etapa de definición. Según nuestro conocimiento no existen trabajos relativos a su validación teórica y empírica. Así mismo, [4] y [16] señalan que la definición de métricas para diagramas que capturen los aspectos dinámicos del software OO es un área relevante, pero un tanto descuidada en el ámbito de la medición del software. Por tal motivo definimos un conjunto de métricas para los diagramas de estados UML [13], algunas miden su complejidad estructural como: *NEntryA* (Número de Acciones de entrada), *NExitA* (Número de Acciones de Salida), *NA* (Número de Actividades), *NSS* (Número de Estados Simples), *NCS* (Número de Estados Compuestos), *NE* (Número de Eventos), *NG* (Número de Guardas), y otras miden su tamaño: *NT* (Número de Transacciones) y *McCabe* (Número ciclomático de McCabe). Mediante la realización de un experimento controlado y su réplica [13], hemos encontrado que *NA*, *NSS*, *NG* y *NT* parecen estar fuertemente correlacionadas con el tiempo empleado en el entendimiento de los diagramas. Esto nos llevó a pensar en la utilidad de la construcción de un modelo de predicción del tiempo de entendimiento de diagramas de estados, basado en los valores de estas métricas. Hemos usado todas las métricas para la construcción del modelo de predicción, ya que consideramos demasiado prematuro el descartar alguna de ellas. Viendo los alentadores resultados obtenidos al aplicar Prototipos Deformables Borrosos [15] para la construcción de modelos de predicción del tiempo de mantenimiento de los diagramas de clases UML [9][10], decidimos usarlos para nuestro propósito.

## **2. Construcción del modelo de predicción del tiempo de entendimiento de los diagramas de estados UML**

Para construir el modelo de predicción, usamos los datos obtenidos en el experimento que se describe en detalle en [13]. Este experimento se llevó a cabo con el objetivo de averiguar si existe correlación entre la complejidad estructural (variable dependiente) y el tiempo de entendimiento de los diagramas de estados (variable independiente).

Los sujetos fueron 11 estudiantes y 8 profesores de la Universidad de Castilla - La Mancha. Los estudiantes estaban matriculados en el último año de Ingeniería Informática y los profesores pertenecían al área de Ingeniería del Software. El experimento consistió en 20 diagramas de estados UML sobre distinto dominio de aplicación, pero lo suficientemente general para ser fácilmente comprendido por los sujetos. Cada diagrama iba acompañado de un test que incluía un cuestionario para evaluar si los sujetos realmente entendían el contenido de cada diagrama de estados. Cada cuestionario contenía cuatro preguntas conceptualmente similares y escritas en el mismo orden, y cada sujeto debía anotar la hora a la que comenzaba a responder el cuestionario y la hora a la que finalizaba. La diferencia entre esas dos anotaciones es lo que llamamos "tiempo de entendimiento" (expresado en segundos).

Por otra parte, para comprobar la validez de este experimento se realizó una réplica del mismo. En este caso, los sujetos eran estudiantes de tercer año de Ingeniería Informática, por lo que la experiencia de los sujetos era menor.

Una vez realizado el experimento, fue necesario transformar los datos obtenidos para que fueran válidos para construir el modelo de predicción del tiempo de entendimiento. Como resultado obtuvimos, filtrando aquellos datos que no cumplieran un cierto nivel de corrección y completitud ( $\geq 0.9$ ), los tiempos de entendimiento mínimo (*TEMin*), promedio (*TEAvg*) y máximo (*TEMax*) para cada diagrama y los datos del cálculo de las métricas de la complejidad estructural de cada diagrama (ver Tabla 1).

D	TEAvg	TEMin	TEMax	NEntryA	NExitA	NA	NSS	NCS	NT	NE	NG	McCabe
1	110,00	15	420	1	1	0	3	0	7	6	2	5
2	95,00	30	170	1	0	3	4	0	7	6	0	4
3	191,94	61	360	2	0	2	4	1	7	4	3	1
4	163,39	69	405	0	0	2	4	0	11	11	2	7
5	129,50	30	215	3	2	2	4	0	13	11	0	9
6	124,56	58	310	6	6	0	6	1	13	12	1	5
7	154,05	72	300	1	0	1	5	2	11	6	3	2
8	140,00	50	360	1	0	3	5	0	13	12	4	9
9	131,79	70	240	0	1	4	5	0	10	7	1	7
10	85,21	50	180	2	1	0	4	0	6	6	0	4
11	153,16	85	360	1	2	1	6	3	17	12	0	3
12	86,37	50	180	1	1	1	3	0	5	5	2	3
13	88,05	35	300	2	1	0	2	0	4	4	0	2
14	136,05	44	360	1	1	2	3	0	8	8	0	5
15	152,22	85	420	1	0	4	9	1	13	11	4	3
16	140,05	50	300	0	0	5	9	0	24	22	1	16
17	108,63	59	195	2	0	1	5	1	8	6	2	2
18	154,89	65	265	2	0	1	12	0	24	23	2	13
19	84,26	40	180	0	1	0	2	0	6	5	0	4
20	85,84	42	140	0	0	0	5	1	12	11	0	7

Tabla 1. Tiempos de entendimiento y valor de las métricas de cada diagrama.

Utilizando los datos que aparecen en la Tabla 1, se construyó el modelo de predicción, siguiendo los pasos que se detallan a continuación.

### 2.1. Obtención de los prototipos usando técnicas de clustering

Con el objetivo de detectar las relaciones entre los diagramas de estados para ser capaces de posteriormente determinar si tienen un tiempo de mantenimiento bajo,

medio o alto, realizaremos un proceso de clustering jerárquico según la técnica de "Emparrillados" (*Repertory Grid*) [1].

Se construyó una matriz de similitud de 20x20 elementos cuyos valores en la diagonal representan los grados de similitud entre los diagramas. Convirtiendo estos valores en porcentajes y ejecutando un algoritmo de clustering jerárquico, obtuvimos los resultados que se muestran en el dendrograma de la figura 1.

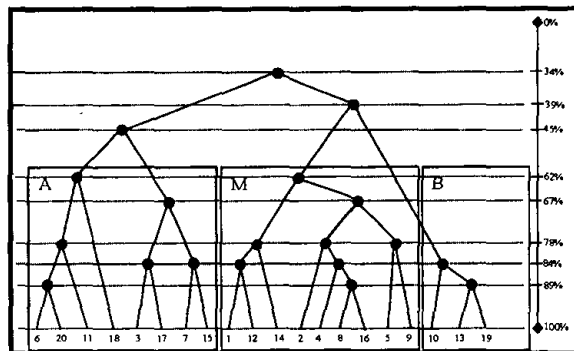


Figura 1. Resultados del Clustering: (B: Tiempo Bajo de Entendimiento, M: Tiempo Medio de Entendimiento, A: Tiempo Alto de Entendimiento).

Una vez obtenidos los resultados del clustering, y con el conocimiento heurístico previo de tener tres prototipos, se realiza un corte a una similitud inferior a un 55%. Con lo cual los diagramas se agrupan en tres prototipos de acuerdo a los valores de las métricas que reflejan su complejidad estructural y tamaño, como muestra la Tabla 2.

Prototipos	Diagramas
B: Tiempo Bajo de Entendimiento	10,13,19
M: Tiempo Medio de Entendimiento	1,2,4,5,8,9, 12,14,16
A: Tiempo Alto de Entendimiento	3,6,7,11,15,17,18,20

Tabla 2. Diagramas agrupados según el prototipo al que pertenecen.

## 2.2. Definición paramétrica de los prototipos

Considerando los prototipos de datos encontrados en la sección anterior y sus valores de tiempos de entendimiento mostrados en la Tabla 1, obtuvimos la definición paramétrica de los prototipos, como muestra la Tabla 3. Los prototipos se definieron por medio de los valores de tiempo de entendimiento medio, máximo y mínimo.

A: T° Alto de Entendimiento		M: T° Medio de Entendimiento		B: T° Bajo de Entendimiento	
Medio	2 min. 15 seg.	Medio	2 min. 5 seg.	Medio	1 min. 25 seg.
Máximo	7 min.	Máximo	7 min.	Máximo	6 min.
Mínimo	42 seg.	Mínimo	15 seg.	Mínimo	35 seg.

**Tabla 3. Definición paramétrica de los prototipos**

**2.3. Representación de los prototipos borrosos**

La utilización de números borrosos triangulares permite que para la representación sea únicamente necesario conocer su centro y el tamaño de la base del triángulo, según se indica en la Tabla 4.

Prototipos	Diagramas	a	Centro	b
B: Tiempo Bajo de Entendimiento	10,13,19	0	0.08	0.74
M: Tiempo Medio de Entendimiento	1,2,4,5,8,9,12,14,16	0	0.26	0.92
A: Tiempo Alto de Entendimiento	3,6,7,11,15,17,18,20	0	0.34	1

**Tabla 4. Definición borrosa de los prototipos.**

**2.4. Deformación de los prototipos borrosos para predecir el tiempo de entendimiento de diagramas de estados.**

Para obtener el valor estimado del tiempo de entendimiento para un nuevo diagrama de estados, los prototipos borrosos se “deforman” considerando el grado de afinidad que tiene este nuevo diagrama con cada prototipo, utilizando el concepto de Prototipos deformables Borrosos [15]. Considerando el diagrama 16 (ver Anexo A), obtuvimos que dicho diagrama tiene un grado de afinidad de 0,712 con el prototipo A y 0,591 con el prototipo M. La Tabla 5 muestra como se obtuvo, a través de una combinación lineal, el tiempo estimado de entendimiento para el diagrama 16.

Media	0,591 / 2	2 min. 5 seg.	+	0,712	=	2 min. 15 seg.
Máximo		7 min.				7 min. 5 seg.
Mínimo		15 seg.				42 seg. 34 s.

**Tabla 5. Combinación lineal modificada para el cálculo de tiempos medios.**

**3. Validación del modelo de predicción**

Para evaluar el poder o la exactitud de predicción del modelo obtenido hemos utilizado las técnicas que se indican en [12]. Utilizando los datos obtenidos en la réplica del experimento, hemos obteniendo como resultados que:

- La magnitud media del error en la réplica (MMRE) es 0.20 y el valor de la mediana (MdMRE) es 0.20, lo que indica que, en promedio, los valores estimados tienen un 80% de exactitud.
- Otro indicador usado es la Predicción a nivel *n*, también conocido como *Pred(n)*, que mide el porcentaje de estimaciones que están dentro de un *n*% de los valores

reales. Según algunas sugerencias [12], 25% es un valor recomendable para  $n$ . Lo que significa que un buen modelo de predicción tendrá una precisión del 75%. En nuestro caso el valor obtenido para  $Pred(25\%)$  es un 70%, lo que indica que el modelo obtenido es válido ya que el 75% de los valores estimados del tiempo de entendimiento tienen al menos un 70% de exactitud. No obstante somos conscientes de que para poder tener mayor certeza de la validez del modelo es necesario validarlo con datos referentes a proyectos reales, tarea que realizaremos en un futuro inmediato.

En el gráfico mostrado en la Figura 2 se pueden observar los valores medios procedentes de la predicción y los valores medios reales.

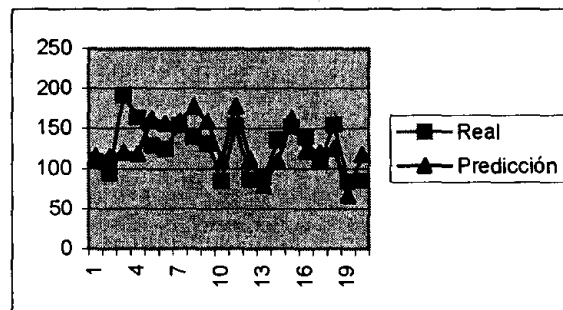


Figura. 2. Valores de la predicción vs. valores reales

#### 4. Conclusiones

En la Ingeniería del Software es ampliamente reconocido que las propiedades estructurales de los diagramas estáticos y dinámicos de UML pueden tener una gran influencia en la calidad del producto software que finalmente se entrega. Por esa razón, la existencia de métricas es crucial, ya que permiten evaluar las propiedades estructurales de una manera cuantitativa y objetiva.

Hemos propuesto la obtención de un modelo de predicción capaz de determinar a priori el esfuerzo de entendimiento que supone cualquier diagrama de estados UML, en función de los valores asociados a las métricas previamente definidas, basado en la utilización de Prototipos Deformables Borrosos [15].

**Agradecimientos.** Este trabajo de investigación es parte del proyecto DOLMEN (TIC ~2000-1673-C06-06) financiado por la Subdirección General de Proyectos de Investigación - Ministerio de Ciencia y Tecnología.



## Referencias

- [1] Bell R. (1990). Analytic Issues in the Use of Repertory Grid Technique. *Advances in Personal Construct Psychology*, 1, pp. 25-48.
- [2] Briand L., Morasca S. and Basili V. (1999). Defining and Validating Measures for Object-Based high-level design. *IEEE Transactions on Software Engineering*, 25(5), pp. 722-743.
- [3] Brito e Abreu F. and Carapuça R. (1994). Object-Oriented Software Engineering: Measuring and controlling the development process. 4th Int Conference on Software Quality, Mc Lean, Va, USA.
- [4] Brito e Abreu F., Zuse H., Sahraoui H. and Melo W. (1999). Quantitative Approaches in Object-Oriented Software Engineering. ECOOP'99 Workshops, LNCS 1743, A. Moreira and S. Demeyer (eds). Springer-Verlag. pp. 326-337.
- [5] Chidamber S. and Kemerer C. (1994). A Metrics Suite for Object Oriented Design, *IEEE Transactions on Software Engineering*. 20(6), pp. 476-493.
- [6] Derr, K. (1995). *Applying OMT*. SIGS Books, Prentice Hall. New York.
- [7] Fenton N. and Pflieger S. (1997). *Software Metrics: A Rigorous Approach*, 2nd. edition. London, Chapman & Hall.
- [8] Genero M., Piattini M., Calero C. (2000). Early Measures For UML class diagrams. *L'Objet*. 6(4), Hermes Science Publications 16(4), pp. 489-515.
- [9] Genero M., Olivas J., Piattini M., and Romero F. (2001). Using metrics to predict OO information systems maintainability., CAISE 2001, Lecture Notes in Computer Science, 2068, Interlaken, Switzerland, 388-401.
- [10] Genero M., Piattini M., Calero C. (2002). An study to validate metrics for class diagrams. *Jornadas Iberoamericanas de Ingenieria de Requisitos y Ambientes de Software (IDEAS'2002)*, La Habana (Cuba), pp. 226-235.
- [11] Marchesi M. (1998). OOA Metrics for the Unified Modeling Language". *Proceedings of the 2nd Euromicro Conference on Software Maintenance and Reengineering*, pp. 67-73.
- [12] Mendes E., Watson I., Triggs C., Mosley N. y Counsell S. (2002). A comparison of development effort estimation techniques for web hypermedia applications. *Eight IEEE Symposium of Software Metrics (Metrics'02)*. IEEE Computer Society Press.
- [13] Miranda D., Genero M. y Piattini M. (2003). Empirical validation of metrics for UML statechart diagrams. *Fifth International Conference on Enterprise Information Systems (ICEIS 03)*, Vol 1., 87-95.
- [14] Object Management Group (1999). UML Revision Task Force, *OMG Unified Modeling Language Specification*, v. 1.3, document ad/99-06-08.
- [15] Olivas J. (2000). *Contribución al Estudio Experimental de la Predicción basada en Categorías Deformables Borrosas*, Tesis Doctoral, Universidad de Castilla La Mancha, España.
- [16] Poels, G. and Dedene, G. (2000). Measures for Assessing Dynamic Complexity Aspects of Object-Oriented Conceptual Schemes. *Proceedings of 19th International Conference on Conceptual Modelling (ER 2000)*, pp. 499-512.
- [17] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W. (1991). *Object-Oriented Modelling and Design*. Prentice Hall. USA.
- [18] Shepperd M., Shofield C. and Kitchenham B. (1996). Effort estimation using analogy. *ICSE 18*, IEEE Computer Society Press.