MEMORIAS

III

SIMPOSIO DE CONTROL AUTOMATICO

MEMORIA

DEL III SIMPOSIO DE CONTROL AUTOMÁTICO

MEMORIA
DEL III SIMPOSIO
DE CONTROL AUTOMÁTICO

Ciudad de La Habana, CUBA
Marzo 19 -23, 2001

EDITOR GENERAL
A. del Pozo Quintero

EDITORES DEL VOLUMEN
Ana C. Campos Hernández
Pedro A. González Lanza

Los trabajos que aparecen en el texto fueron reproducidos usando los manuscritos enviados por los autores, que utilizaron diferentes letras y procesadores de textos. Por esta razón, las figuras y tablas de algunos trabajos no coinciden exactamente con los requerimientos solicitados en el llamado inicial, y consecuentemente la reproducción no muestra una completa uniformidad. Para asegurar una publicación rápida, esas inconsistencias no pudieron ser cambiadas por falta de tiempo. Pedimos disculpas a los lectores por cualesquiera deficiencias que puedan ser encontrada en este volumen.

Instituto de Cibernética, Matemática y Física
Calle 15 No. 551 e/ C y D, Vedado, Plaza de la Revolución
Ciudad de La Habana, Cuba
Telf. (537) 32-0319 Fax(537) 33-3373

Editores: Abelardo del Pozo Quintero
Ana C. Campos Hernández
Pedro A. González Lanza

ISBN: 959-7056-12-7

PROLOGO

Los organizadores de eventos en la esfera del Control Automático, a través de la historia, han tratado de mantener una relación estrecha en la presentación de trabajos que cubran la teoría y la práctica. Esto ha sido reforzado por las actividades de la Federación Internacional de Control Automático (IFAC), mediante sus congresos técnicos cada tres años y la constitución de sus comités de teoría y aplicación. Otras Organizaciones Internacionales también convocan a eventos de Control Automático, incluyendo ambos aspectos.

El Tercer Simposio Internacional de Control Automático dentro del marco de CIMAF 2001, tratara de mantener esta relación en los diferentes tópicos convocados, con la discusión de los trabajos teóricos y aplicados de los participantes, provenientes de 8 países (5 de América y 3 de Europa)

Se aprobaron por el Comité Internacional de Programa un total de 30 trabajos, de los cuales, 8 tienen un carácter teórico, que abarcan desde nuevos enfoques de cuestiones clasicas del control convencional hasta aspectos de la inteligencia artificial aplicada al control. Por otro lado, 18 trabajos cubren las contribuciones de los participantes en aplicaciones de la teoría en una gama de áreas tan disímiles como son: agricultura, salud, educación, industria química, calderas, motores eléctricos y robótica. Se han separado en forma de conferencias 4 trabajos, que tratan de dar una visión un poco más amplia que una ponencia de aspectos interesantes del control. Por último, la Red de Automática de Cuba (RAC), visto el parecer del Comité Científico del evento, publicara una selección de los mejores trabajos en su órgano oficial: la Revista de Electrónica, Automática y Comunicaciones.

Es un placer expresar nuestro agradecimiento al Comité Internacional de Programa, por su apoyo en la preparación y selección de los trabajos presentados a este Simposio. Adicionalmente, queremos dejar plasmado en estas líneas, nuestro agradecimiento a la compañera Nadiezhda Torralba, sin cuya constante y eficaz ayuda en la comunicación continua con los participantes y en la recepción de los trabajos y su envio a los evaluadores, asi como en los más diversos aspectos organizativos, hubiera sido imposible la concreción de este evento.

El objetivo principal de los organizadores de este Simposio, que se celebra cada dos años, es que investigadores y demas profesionales que trabajan en este campo, salgan satisfechos después de reunirse, discutir y confrontar con otros especialistas de diferentes latitudes sus resultados, y a la vez que sirva para elevar cada vez más los conocimientos teóricos y prácticos de los asistentes.

Finalmente, les deseamos a todos el mayor de los éxitos y una grata estancia en nuestro país.

Dr. Abelardo del Pozo Quintero
Presidente del III Simposio de Control Automático

Comité Organizador CIMAF2001

Presidenta de Honor: Dra. Rosa E. Simeon Negrin
Ministra de Ciencia, Tecnologia y Medio Ambiente de la Republica de Cuba.

Presidente: Dr. Manuel Lazo Cortes.
Director del Instituto de Cibernetica, Matemática y Fisica. (ICIMAF)

III Simposio de Control Automático

Comité Internacional de Programa

Dr. Abelardo del Pozo Quintero. Cuba.
Dra. Ana C. Campos Hernández. Cuba.
Dr. Ernesto Cornieles. Venezuela.
Dr. Jesus M. Zamarreño Cosme. España.
Dr. Ieroslam Baruch. Bulgaria.
Dra. Maria J. de la Fuente Aparicio. España.
Dr. Orestes Llanes Santiago Cuba.
Dr. Vicente Feliu. España
Dr. Raul Rivas Perez. Cuba.
Dr. Alberto Aguado Behar. Cuba.
Dr. José R. Peran González. España
MSc. Pedro A. González Lanza. Cuba
Ing. Alfredo Gomez Infante. Cuba

de la

MEMORIA III SIMPOSIO DE CONTROL AUTOMATICO 2001

INDICE

Adaptive Prototypes: A Tool for Prediction. A Case Study: UML Class Diagram Maintainability

José A. Olivas, Marcela Genero, Francisco P. Romero, Mario Piattini
Department of Computer Science, University of Castilla-La Mancha
Ronda de Calatrava 5, 13071, Ciudad Real (Spain)
E-mail: {jaolivas, mgenero, fpromero, mpiattin}@inf-cr.uclm.es

Abstract

It is presented an application of Adaptive Prototypes Theory (Olivas 2000b) and arrive at a method for predicting UML class diagram maintainability. This method then allows us to interpret a real situation using a previously established paradigm and define the current situation. The use of fuzzy logic provides a new frame for this prediction model.

Taking into account that the quality of object oriented information systems (OOIS) depends greatly on the decisions taken at early phases of their development, class diagrams lay the foundation for all later design work. So, their quality heavily affects the product that will be ultimately implemented.

The aim of this paper is to predict UML Class Diagram Maintainability using metrics measured in early phases. We will start with a FPKD process (Fuzzy Prototypical Knowledge Discovery) for finding Fuzzy Adaptive Prototypes of maintainability, and later we will predict a real case maintainability deforming the similar prototypes using the degree of compatibility with them.

Keywords. fuzzy deformable prototypes, FPKD, prediction models, object oriented information systems maintainability.

1. Fuzzy Prototypes and FPKD process.

1.1. Concepts and Prototypes.

Fayyad and his collaborators (Fayyad 1996) define KDD process like "the non-trivial process to identify valid, new, potentially useful and comprehensible patterns in data". Using multidiscipline techniques and seeing as they act together. The term process implies that there are several steps, like the preparation of data or the search for patterns. The term pattern talks about a new set of data, that must be valid for new data with some degree of certainty. These patterns must be comprehensible, if it is not immediately, after a postprocessing, This definition implies that we have to consider measures of certainty (capacity of classification of new data) or utility (quality of the predictions of the basis of theses prototypes of data).

Taking the prototype theory from the cognitive psychology as a reference, a single representation could be seen as prototypical. However, in a previous approximation at the Knowledge Acquisition process we were able to observe that this representation excessively simplifies the behavioural guidelines of the experts. When a technician is confronted with a real situation, he/she handles a ranges of prototypes determined by a series of factors and must decide which type of maintainability is to be expected. Therefore, the prototype "Maintainability" is not unique.

Zadeh's mentions the classical prototype theories form the point of view of cognitive psychology, criticizing precisely what we have just pointed out: That these theories do not fit the function that a prototype should do. Zadeh's approach to what must be taken as a prototype is less intuitive than the conceptions of psychological theories but is more rational and closer to the meaning of a prototypical concept displayed in a more detailed examination. In this case, we have observed that Zadeh's idea suggests a concept that encompasses a set of prototypes. Which represent the high, medium, or low compatibly of the samples with the concept. "The prototype is not a single object or even a group of objects in A. Rather, it is a fuzzy schema for generating a set of objects which is roughly coextensive with A", being A the Concept or Predicate (Zadeh, 1982).

For our purposes, it will be convenient to stratify the representation of A by grouping together elements which have the same, or nearly the same, grades of membership, and quantifying the grades into a small number of levels, three or four. Each one of the groups of exemplars in A is processed separately.

1.2. The FPKD Process.

Modifications to the original KDD process are proposed, as it represents fig 1. which they involve incorporation of new knowledge in different points and decisions of the users or experts. The aim must be to generate conceptual prototypes that allow to evaluate new situation from these patterns, and to establish predictions if these patterns represent temporary cycles. The stages of the modified KDD (from now on Fuzzy Prototypical Knowledge Discovery: FPKD) are the following ones (fig 1).

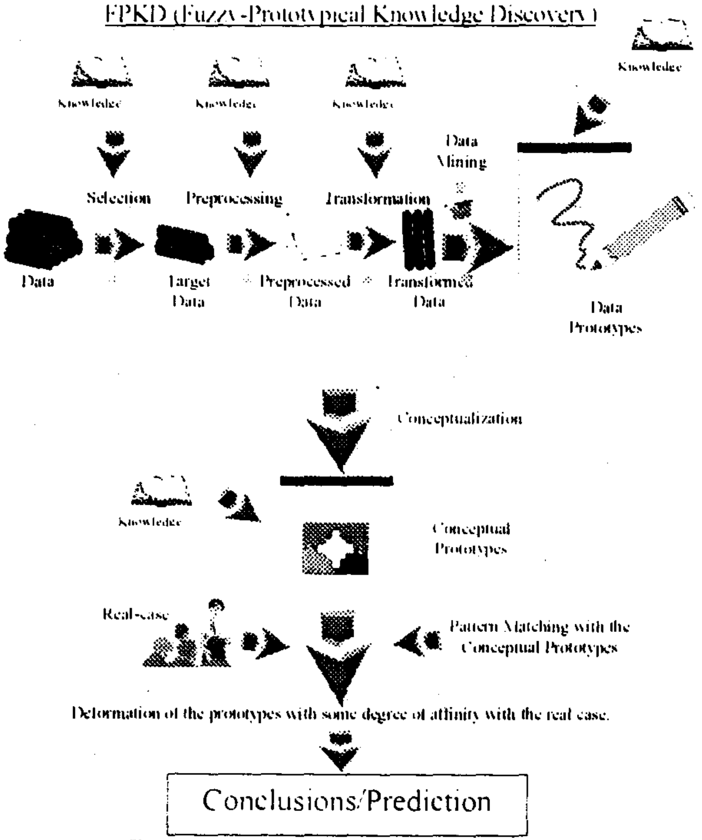


Figure 1. FPKD and Prediction Processes.

2. A comprehensive controlled experiment to build a prediction model for UML class diagram maintainability

A widely accepted principle in software engineering is that the quality of a software product should be assured in the early phases of its life cycle. In a typical OOIS design at the early phases, a class diagram is first built. The class diagram is not merely the basis of modelling the persistent system data. In OO modelling, where data and process are closely linked, class diagrams provide the solid foundation for the design and implementation of OOIS.

As an early available. key analysis artifact the quality of the class diagram is crucial to the success of system development. Generally. problems in the artifacts produced in the initial phases of system development propagate to the artifacts produced in later stages. where they are much more costly to identify and correct (Boehm, 1981). As a result. improving the quality of class diagrams. will therefore be a major step towards the quality improvement of the OOIS development. The appearance of UML (Object Management Group. 1999). as standard OO modelling language. should contribute to this. Despite this. we have to be aware that a standard modelling language can only give us syntax and semantics to work with. but it cannot tell us whether a "good" model has been produced. Naturally. even when language is mastered. there is no guarantee that the models produced will be good. Therefore. it is necessary to assess their quality.

The definition of the different characteristics that compose the concept of "quality" is not enough on its own in order to ensure quality in practice, as people will generally make different interpretations of the same concept. Software measurement plays an important role in this sense because metrics provide a valuable and objective insight into specific ways of enhancing each of the software quality characteristics. Measurement data can be gathered and analysed to assess current product quality. to predict future quality. and to drive quality improvement initiatives (Tian. 1999). Quality is a multidimensional concept, composed of different characteristics such as functionality. reliability. usability. efficiency. maintainability and portability (ISO. 1999). These characteristics are external quality attributes that can only be measured late in the OOIS life cycle. Therefore it is necessary to find early indicators of such qualities based. for example. on the structural properties of class diagrams (Briand, 1999b).

Within the field of software engineering a plethora of metrics have been proposed for measuring OO software products, even though most of them are related to products obtained from advanced design and implementation phases (Chidamber and Kemerer. 1994. Lorenz and Kidd. 1994: Brito e Abreu and Melo. 1996; Henderson-Sellers. 1996; Fenton and Pflegeer. 1997). Few studies have been made specifically about measures applied to UML class diagrams (Marchesi. 1998; Genero et al.. 2000). and they lack validation that would assure that they really are fruitful in practice. For our purpose we distinguish the following maintainability sub-characteristics:

- Understandability. The ease with which the class diagram can be understood.
- Analysability. The capability of the class diagram to be diagnosed for deficiencies or to identify parts to be modified.
- Modifiability. The capability of the class diagram to enable a specified modification to be implemented.

Our motivations are to present metrics for measuring UML class diagram structural complexity (internal quality attribute) and secondly demonstrate through experimentation that it can be used to predict UML class diagram maintainability (external quality attribute). which will strongly influence OOIS maintainability.

2.1. UML Class Diagram Metrics.

After performing a thorough review of several OO metric proposals (Chidamber and Kemerer. 1994; Lorenz and Kidd. 1994. Brito e Abreu and Melo. 1996. Marchesi. 1998. Genero et al. 2000) that can be applied at early phases of the OOIS development. we will only present here those that can be applied to the class diagram as a whole. calling them "Class Diagram-Scope metrics". We classify them in two categories: open-ended metrics. whose values are not bounded in an interval. and close-ended metrics whose values are bounded. in our case in the interval [0.1].

Close-Ended metrics

- Number of Classes. (NC) is the total number of classes within a class diagram.

- NUMBER OF ATTRIBUTES. (NA) is the total number of attributes within a class diagram.
- NUMBER OF METHODS. (NM) is the total number of methods within a class diagram.
- NUMBER OF ASSOCIATIONS. (NAssoc) is defined as the total number of associations within a class diagram.
- NUMBER OF AGGREGATION. (NAgg) is defined as the total number of aggregation relationships within a class diagram (each whole-part pair in an aggregation relationship).
- NUMBER OF DEPENDENCIES. (NDep) is defined as the total number of dependencies relationship within a class diagram.
- NUMBER OF GENERALISATIONS. (NGen) is defined as the total number of generalisation relationships within a class diagram (each parent-child pair in a generalisation relationship).
- NUMBER OF GENERALISATIONS HIERARCHIES. (NGenH) is defined as the total number of generalisations hierarchies in a class diagram
- MAXIMUM DIT. The Maximum DIT in a class diagram is the maximum between the DIT value obtained for each class of the class diagram. The DIT value for a class within a generalisation hierarchy is the longest path from the class to the root of the hierarchy.

Open-ended Metrics

- NUMBER OF ASSOCIATIONS VS. CLASSES. (NAssocVC) is defined as the ratio between the number of associations in a class diagram (NAssoc) divided by the total number of classes in the class diagram (NC).
- NUMBER OF DEPENDENCIES VS. CLASSES. (NDepVC) is defined as the ratio between the number of dependencies in a class diagram (NDep) divided by the total number of classes in the class diagram (NC).
- NUMBER OF AGGREGATIONS VS. CLASSES. (NAggVC) is defined as the ratio between the number of aggregations in a class diagram (NAgg) divided by the total number of classes in the class diagram (NC).
- NUMBER OF GENERALISATIONS VS. CLASSES. (NGenVC) is defined as the ratio between the number of generalisations in a class diagram (NGen) divided by the total number of classes in the class diagram (NC).

Taking into account some suggestions provided in Briand et al. (1999b; 1999c) about how to do empirical studies in software engineering, we carried out a controlled experiment with the goal of predicting UML class diagrams maintainability from metric values obtained at the early phases of OOIS life cycle.

2.2. Experiment.

The experimental subjects used in this study were: 7 professors and 10 students enrolled in the final-year of Computer Science at the University of Castilla-La Mancha in Spain. All of the teachers belong to the Software Engineering area and they have enough experience in the design and development of OO software. By the time the experiment was done all of the students who had two courses on Software Engineering, in which they learnt in depth how to build OO software using UML. Moreover, subjects were given an intensive training session before the experiment took place.

The subjects were given twenty eight UML class diagrams of the same universe of discourse, related to Bank Information Systems. Each diagram has a test enclosed which includes the description of maintainability sub-characteristics, such as: understandability, analysability, modifiability. Each subject has to rate each sub-characteristic using a scale consisting of seven linguistic labels. For example for understandability we proposed the following linguistic labels:

Extremely difficult to understand	Very difficult to understand	A bit difficult to understand	Neither difficult nor easy to understand	Quite easy to understand	Very easy to understand	Extremely easy to understand

We allowed one week to do the experiment, i.e., each subject had carry out the test alone, and could use unlimited time to solve it.

2.3. Construction of fuzzy deformable prototypes to characterise UML class diagram maintainability

We have used an extension of the traditional Knowledge Discovery in Databases (KDD) (Fayyad, 1996): the Fuzzy Prototypical Knowledge Discovery (FPKD) that consists of the search for fuzzy prototypes (Zadeh, 1982) that characterise the maintainability of an UML class diagram. In the rest of this section we will explain each of the steps we have followed in the FPKD (see figure 1).

1. *Selection: the target data*

We have taken as a start set a relational database that contains 476 records (with 16 fields,13 represent metrics values, 3 represent maintainability sub-characteristics) obtained from the calculation of the metric values (for each class diagram) and the responses of the experiment given by the subjects.

2. *Preprocessing*

The Data-Cleaning was not necessary because we didn't find any errors.

3. *Transformation*

This step was performed doing different tasks:

- SUMMARISING SUBJECT RESPONSES. We built a unique table with 28 records (one record for each class diagram) and 17 fields (13 metrics and 3 maintainability sub-characteristics). This table is shown in Appendix A). The metric values were calculated measuring each diagram, and the values for each maintainability sub-characteristics were obtained aggregating subjects's rating using the mean of them.
- CLUSTERING BY REPERTORY GRIDS. In order to detect the relationships between the class diagrams, for obtaining those which are easy, medium or difficult to maintain (based on subject rates of each maintainability sub-characteristics), we have carried out a hierarchical clustering process by Repertory Grids. The set of elements is constituted by the 28 class diagrams, the constructions are the intervals of values of the subjects'rating. To accomplish an analysis of clusters on elements, we have built a proximity matrix that represents the different similarities of the elements, a matrix of 28 x 28 elements (the diagrams) that above the diagonal represents the distances between the different cycles. Converting these values to percentages, a new table is created and the application of Repertory Grids Analysis Algorithm returns a graphic as a final result (see figure 2).

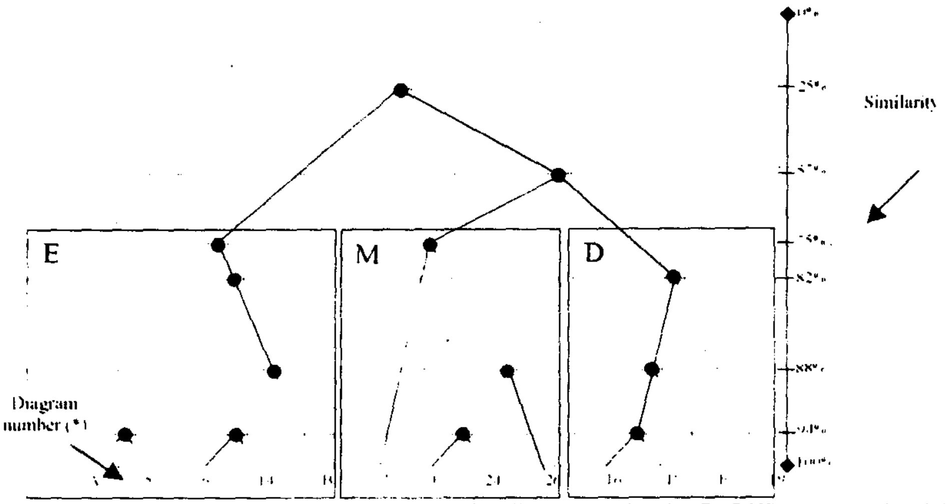


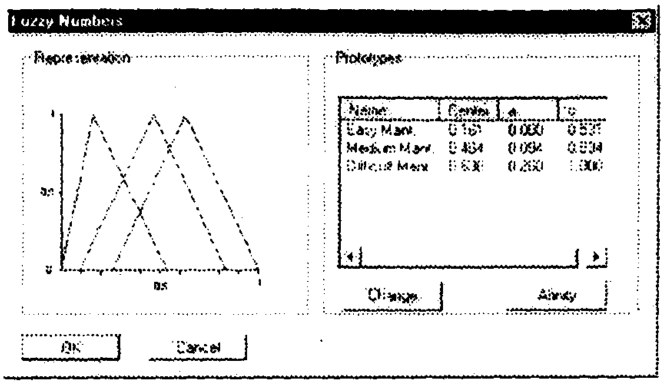
Figure 2. Clustering results (E: Easy to maintain, M: Medium to maintain, D: Difficult to maintain) (*) We have grouped some class diagrams assigning them one letter because they have 100% of similarity (see appendix A)

- DATA MINING. The selected algorithm for data mining process was summarise functions. Table 1 shows the parametric definition of the prototypes. These parameters will be modified taking into account the degree of affinity of a new class diagram with the prototypes. With the new modified prototype we will be able to predict the maintainability of a new class diagram.

Difficult	Understandability	Analisability	Modifiability
Average	6	6	6
Max.	6	6	7
Min.	6	5	6
Medium			
Average	5	5	5
Max.	5	6	5
Min.	4	4	4
Easy			
Average	2	2	3
Max.	3	3	3
Min.	2	2	2

Table 1. Prototypes "Easy, Medium and Difficult to maintain"

- FORMAL REPRESENTATION OF CONCEPTUAL PROTOTYPES. The prototypes have been represented as fuzzy numbers, which are going to allow us to obtain a degree of membership in the concept. For the sake of simplicity in the model, they have been represented by triangular fuzzy numbers. Therefore, in order to construct the prototypes (triangular fuzzy numbers) we only need to know their centerpoints ("center of the prototype"), which are obtained by normalising and aggregating the metric values corresponding to the class diagrams of each of the prototypes (see figure 3).



Name	Center	a	c
Easy Mant.	0.151	0.000	0.531
Medium Mant.	0.464	0.094	0.834
Difficult Mant.	0.630	0.260	1.000

Figure 3. Representation of the prototypes

2.4. Prediction of UML class diagram maintainability.

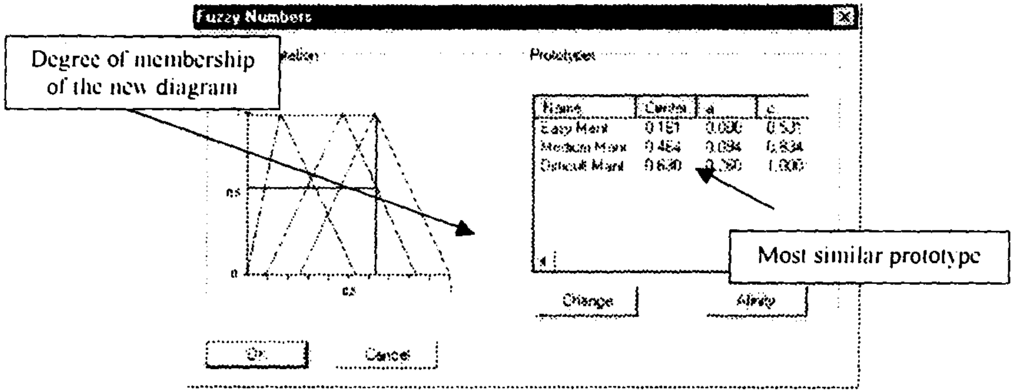
Using Fuzzy Deformable Prototypes (Olivas, 2000a; 2000b), we can deform the most similar prototype to a new class diagram, and define the factors for a new situation, using a linear combination with the degrees of membership as coefficients. We will show an example of how to deform the fuzzy prototypes found in section 3.5. Given the following metric values corresponding to a new class diagram:

NC	NA	NM	NAssocR	NAssocVC	NAggR	NAggRVC	NAggH	NdepR	NdepVC	NGenR	NGH	MaxDIT
21	30	70	10	0.47619048	6	0.28571429	2	3	0.14285714	20	5	2

And their normalized values:

NC	NA	NM	NAssocR	NAssocVC	NAggR	NaggRVC	NAggH	NDepR	NDepVC	NGenR	NGH	MaxDIT
0.70	0.50	0,69	0,71	0,48	0,67	0,45	0,67	0,75	0,43	0,83	1,00	0,40

The final average is 0,64. The affinity with the prototypes is shown in figure 4.



Name	Center	a	c
Easy Mant	0.151	0.000	0.531
Medium Mant	0.464	0.094	0.834
Difficult Mant	0.630	0.260	1.000

Figure 4. Affinity of the real case with the prototypes

The most similar prototype for this new class diagram is "Difficult to maintain", with a degree of membership of 0.98. Then, the prediction is:

	Understandability	Analyzability	Modifiability
Average	6	6	6
Maximum	6	6	7
Minimum	6	5	6

We want to highlight that this a first approach to predict UML class diagram maintainability. we need "real data" about UML class diagram maintainability efforts. like time spent in maintenance tasks in order to predict data that can be highly useful to software designers and developers.

3. Conclusions and future work.

We have carried out a controlled experiment. with the objective of predicting UML class diagram maintainability based on the metrics values and the expert's rating of each of the maintainability sub-characteristics. The prediction model is an extension of the traditional KDD called FPKD and a novel technique which can be used for prediction based on Fuzzy Deformable Prototypes (Olivas. 2000a; 2000b). This model have been used for different kinds of real problems. such as forest fire prediction. financial analysis or medical diagnosis. with very good results. as in this case study.

In future work. we will focus our research on predicting other quality factors like those proposed in the ISO 9126 (1999). which not only tackle class diagrams. but also evaluate other UML diagrams. such as use-case diagrams. E/R diagrams. state diagrams. etc.

Acknowledgements

This research is part of the MANTICA project. partially supported by CICYT and the European Union (1FD97-0168).

References

1. Basili V.. Shull F. and Lanubile F. Building knowledge through families of experiments. *IEEE Transactions on Software Engineering*. 25(4). 435-437. 1999.
2. Boehm. B. *Software Engineering Economics*. Prentice-Hall (1981).
3. Briand L.. Morasca S. and Basili V. Defining and Validating Measures for Object-Based high-level design. *IEEE Transactions on Software Engineering*. 25(5). 722-743. 1999a.
4. Briand L.. Arisholm S.. Counsell F.. Houdek F.. and Thévenod-Fosse P. Empirical Studies of Object-Oriented Artifacts. Methods. and Processes: State of the Art and Future Directions. *Technical Report IESE 037.99/E. Fraunhofer Institute for Experimental Software Engineering*. Kaiserslautern. Germany. 1999b.
5. Briand L.. Bunse C. and Daly J. A Controlled Experiment for evaluating Quality Guidelines on the Maintainability of Object-Oriented Designs. *Technical Report IESE 002.99/E. Fraunhofer Institute for Experimental Software Engineering*. Kaiserslautern. Germany. 1999c.
6. Briand L.. Wüst J.. Daly J. and Porter D. Exploring the relationships between design measures and software quality in object-oriented systems. *The Journal of Systems and Software 51*. 245-273. 2000.
7. Brito e Abreu F. and Melo W. Evaluating the Impact of Object-Oriented Design on Software Quality. *Proceedings of 3^{d} International Metric Symposium*. 1996.
8. Brito e Abreu F.. Zuse H.. Sahraoui H. and Melo W. Quantitative Approaches in Object-Oriented Software Engineering. *Object-Oriented technology: ECOOP'99 Workshop Reader*. Lecture Notes in Computer Science 1743. Springer-Verlag. 326-337. 1999.
9. Chidamber S. and Kemerer C. A Metrics Suite for Object Oriented Design. *IEEE Transactions on Software Engineering*. 20(6). 476-493. 1994.

10. Daly J., Brooks A., Miller J., Roper M. and Wood M. Evaluating Inheritance Depth on the Maintainability of Object-Oriented Software. *Empirical Software Engineering*, 1. Kluwer Academic Publishers, Boston, 109-132, 1996.
11. Derr K. *Applying OMT.* SIGS Books, New York, 1995.
12. Fayyad U., Piatetsky-Shapiro G. and Smyth P. The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Communications of the ACM*, 39(11), 27 – 34, 1996.
13. Fenton N. and Pfleeger S. *Software Metrics: A Rigorous Approach*. 2nd. edition. London, Chapman & Hall, 1997.
14. Genero, M., Piattini, M. and Calero, C. Early Measures For UML class diagrams. *L'Objet*. 6(4), Hermes Science Publications, 2000, (to appear).
15. ISO/IEC 9126-1.2. Information technology- Software product quality – Part 1: Quality model. 1999.
16. Henderson-Sellers B. *Object-Oriented Metrics - Measures of complexity*. Prentice-Hall, Upper Saddle River, New Jersey, 1996.
17. Lorenz M. and Kidd J. *Object-Oriented Software Metrics: A Practical Guide*. Prentice Hall, Englewood Cliffs, New Jersey, 1994.
18. Marchesi M. OOA Metrics for the Unified Modeling Language. *Proceedings of the 2nd Euromicro Conference on Software Maintenance and Reengineering*, 67-73, 1998.
19. Object Management Group. *UML Revision Task Force. OMG Unified Modeling Language Specification, v. 1.3, document ad/99-06-08*, 1999.
20. Olivas J. A. and Romero F. P. FPKD. Fuzzy Prototypical Knowledge Discovery. Application to Forest Fire Prediction. *Proceedings of the SEKE'2000*, Knowledge Systems Institute, Chicago, Ill. USA, 47 – 54, 2000a.
21. Olivas J. A. Contribution to the Experimental Study of the Prediction based on Fuzzy Deformable Categories, PhD Thesis, University of Castilla-La Mancha, Spain, 2000b.
22. Poels G. On the use of a Segmentally Additive Proximity Structure to Measure Object Class Life Cycle Complexity. *Software Measurement : Current Trends in Research and Practice*, Deutscher Universitäts Verlag, 61-79, 1999.
23. Poels G. On the Measurement of Event-Based Object-Oriented Conceptual Models. *4th International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering*, June 13, Cannes, France, 2000.
24. Tian J. Taxonomy and Selection of Quality Measurements and Models. *Proceedings of SEKE'99, The 11th International Conference on Software Engineering & Knowledge Engineering*, June 16-19, 71-75, 1999.
25. Zadeh, L. A.A note on prototype set theory and fuzzy sets. *Cognition* 12, 291 – 297, 1982.
26. H. Zuse. *A Framework of Software Measurement*. Berlin, Walter de Gruyter, 1998.

Appendix A

The following table shows in each row the number of the class diagrams used in the experiment described in section 3, and in each column their metric values. Attached to some diagrams appear one letter. The diagrams which have the same letter mean that they have 100% of similarity.

	NC	NA	NM	NAssoc	NAssocVC	NAgg	NAggVC	NAggH	NDep	NDepVC	NGen	NGH	MaxDIT	Understandability	Analyzability	Modifiability
D0	2	4	8	1	0.5	0	0	0	0	0	0	0	0	1	1	1
D1 (A)	3	6	12	1	0,33	1	0.33	1	0	0	0	0	0	2	2	2
D2 (A)	4	9	15	1	0,25	2	0.5	1	0	0	0	0	0	2	2	2
D3 (A)	3	7	12	3	1	0	0	0	0	0	0	0	0	2	2	2
D4 (A)	5	14	21	1	0,2	3	0,6	2	0	0	0	0	0	2	2	2
D5	3	6	12	2	0.66	0	0	0	0	0	0	0	0	2	2	2
D6	4	8	12	3	0.75	0	0	0	1	0,25	0	0	0	2	3	3
D7 (B)	6	10	14	2	0.33	2	0.33	1	0	0	2	1	1	3	3	3
D8 (A)	3	9	12	1	0,33	0	0	0	1	0,33	0	0	0	2	2	2
D9 (B)	7	14	20	2	0.28	3	0,42	1	0	0	2	1	1	3	3	3
D10 (B)	9	18	26	2	0,22	3	0,33	1	0	0	4	2	1	3	3	3
D11 (B)	7	18	37	3	0.42	3	0,42	1	0	0	2	1	1	3	3	3
D12 (B)	8	22	35	3	0,375	2	0,25	1	1	0,125	2	1	1	3	3	3
D13 (A)	5	9	26	0	0	0	0	0	0	0	4	1	2	2	2	2
D14	8	12	30	0	0	0	0	0	0	0	10	1	3	2	3	3
D15 (C)	11	17	38	0	0	0	0	0	0	0	18	1	4	4	4	4
D16	20	42	76	10	0,5	6	0,3	2	2	0,1	10	3	2	6	6	6
D17 (D)	23	41	88	10	0,43	6	0,23	2	2	0,06	16	3	3	6	6	6
D18 (E)	21	45	94	6	0,28	6	0,28	2	1	0,04	20	2	4	6	5	6
D19	29	56	98	12	0,41	7	0,24	3	3	0,1	24	4	4	6	6	7
D20 (B)	9	28	47	1	0,11	5	0,55	2	0	0	2	1	1	3	3	3
D21 (F)	18	30	65	3	0,166	5	0,277	1	0	0	19	2	4	5	5	5
D22 (D)	26	44	79	11	0,42	6	0,23	2	0	0	21	5	3	6	6	6
D23 (F)	17	32	69	1	0,05	5	0,19	1	0	0	19	1	5	5	5	5
D24	23	50	73	9	0,4	7	0,3	3	2	0,08	11	4	1	5	6	5
D25 (E)	22	42	84	14	0,63	4	0,18	2	4	0,18	16	3	3	6	5	6
D26	14	34	77	4	0,28	9	0,64	2	0	0	7	2	4	4	5	5
D27 (C)	17	34	47	6	0,35	6	0,35	3	0	0	11	2	2	4	4	4

ISBN 959-7056-12-7
9 789597 056126

CIMAF 2001

CONFERENCIA INTERNACIONAL

III SIMPOSIO DE CONTROL AUTOMÁTICO

Se le otorga a:

José A. Olivas, Marcela Genero, Francisco P. Romero, Mario Piattin

Por ser autores del trabajo:

Adaptive Prototypes: A Tool for Prediction. A Case Study: UML Class Diagram Maintainability

Dado en Ciudad de La Habana

Día 23 Mes Marzo Año 2001

CIMAF

Dr. Abelardo del Pozo Quintero
Presidente Comité Organizador