INTEGRATED ENVIRONMENT OF SYSTEMS AUTOMATED ENGINEERING.

Autores
Arturo Carlos SERVETTO, Ramón GARCÍA MARTÍNEZ, Gregorio PERICHINSKY
aserve/rgm/gperi@mara.fi.uba.ar
Tel. (011) 4342-9184 / 4343-0891, Int. 142
Laboratorio de Bases de Datos y Sistemas Operativos
Facultad de Ingeniería
Universidad de Buenos Aires

KEYWORDS: static models, dynamic models, integrated environment, conceptual pattern.

ABSTRACT.

The Standardization of a development process for systems of medium complexity, entirely supported by an integrated development environment, could mean a double contribution to the industry of systems development. On the one hand it would contribute to the reduction of costs by shortening times of development and facilitating the maintenance of the systems. On the other hand, by using conceptual models of a high level of abstraction, final users would be able to visualize and follow the progress of systems development and to commit themselves in the process actively, guaranteeing their success.

The purpose of this series of papers is to identify, design, develop and integrate the components of an integrated environment for a system automated development, starting from high-level-abstraction formal specifications. It is intended to achieve a generation of systems starting from only two models: the static or data structure model, and the dynamic or functional model. The former is based on an adaptation of the conceptual pattern of entities and relationships, and the latter on the formal specification of operations in objects relational algebra and on the finite automaton theory. The maintenance of the systems generated by the tool would be made by operating directly on the static and dynamic models, with no need for either re-coding or making reverse engineering.

The strength of the proposal is based on integration of the Software Engineering and the analysis and metrics of quality. Storage protocols of data definitions and storage protocols of interface definitions corresponding to a graph, through the state transitions and their gradation so that the user could visualize the development of the life cycle[6][32][33][47].

Any artifact or document can be measured and estimated if it is produced during the software life cycle.

This study is an approximation to a process of activities supported by an effective help tool. Abstraction and visualization capacities are provided to the artifacts with the system design and analysis (CASE). Such artifacts under controlled conditions trace the software life cycle taking information of the structure (framework) of the application; with Quality, Reliability and Metrics.

PROCESS ORIENTED.

Similarly to what happened with the imperative programming and the structured systems development, at the beginning the Object Oriented (O-O) paradigm was applied exclusively to the programming. The following phase, more or less in the 80s, was characterized by the interest explosion in the interfaces of the user. The most evident effect is reflected in the current abundance of libraries of classes for the development of interfaces[24][36][38].

The current phase of the history of the O-O is characterized by the emphasis which has been moved from programming to analysis and design, and because there is conscience of the problem of the open systems and of the need for standards. There exists an important trend toward the incorporation of methods guided to objects, in the systems of database management as well as in the structured methods of existing management and the CASE tools that give support to systems of database and structured methods[2]. The proposal of this work consists in the specification of a process and behavior model of systems for a CASE tool that, based on the analysis by scenarios and in the classification of objects in a system in objects of application and of interface, synthesizes the

functionality and the behavior of a system in a finite states machine by scenery [1][9][11][17] [18][25] [26][34][35][37][39][45][46][53] .

The states in the model represent objects of interface and the transitions represent messages of activation to other objects of interface or well messages to classes of application that give as a result a set of objects that is associated with an object interface. At the same time each object interface, associated with an empty set, unitary or vast of objects of application, it can have some of these active objects or in area (inclusive null or undefined object).

Storage protocols of data definitions and the definitions grammar corresponding to a graph can be simple (defined attribute, aggregate entity, aggregate attribute, identifier and hierarchy) or composites (attributes, entities and relations). Another protocol are the storage protocols of interface definitions and the definitions grammar corresponding to a graph can be simple (report, attribute value, computed value, conjunction term and code) or composites (intefaces, forms, transitions, relations between forms and a disjunction term).

An adjustment is used for the formal specification of the messages to classes of application of relational algebra, whose operations determine instance subsets of classes of application or joint of instances of temporary classes, created as a result of consulting. The model results apt for the automation of the office-supply systems development or from traditional management and reduces the behavioral specification and functionality of a system to an only graph by all scenarios, synthesizing the scenarios or cases of use, the interaction among objects (sequence graphs and of collaboration) and the state sequences, proposed by the UML (Unified Modeling Language) of Booch, Rumbaugh and Jacobson.

REFERENCES

[1] Anfossi, D; Servetto, A., 1997, Review of Object Oriented CASE tools: Comparisons, Evaluation and Conclussion; proceeding of the III ICIE (International Conference on Informatics Engineering). University of Buenos Aires; pp. 449-463. Spanish.

[2] Bandinelli, S. 1995. Modelling and Improving an Industrial Software Process. IEEE Transaction on Software Engineering. Vol.21, N°5, Pág. 440-454.

[3] Bass, L. , Clements, P. and Kazman, R. "Software Architecture in Practice". Addison - Wesley. 1998.

[4] Bell, D. and Morrey, I. "Software Engineering, Second Edition".Prentice Hall.1992.

[5] Böehm 1981 B. Boehm, Software Engineering Economics, Prentice Hall, 1981.

[6] Böehm, B. 1988. A Spiral Model for Software Development an Enhacement. IEEE Computer. Mayo, Pág. 61-72.

[7] Brown, A. W. 1989, Database Support for Software Engineering. New York: Wiley.

[8] Brown, A. W. & McDermid, J. A., 1991, On Integration and Reuse in a Software Development Environment, F. Long & M. Tedd (Editors), Ellis Horwood, Mar 7.

[9] Chappell, C., Downes, V. & Tully, T., 1989, Real-Time CASE: The Integration Battle, Ovum.

[10] Constantine, L. and Lockwood, L. "Software for use". A Practical Guide to the Models and Methods of Usage - Centred Design. Addison - Wesley. 1999.

[11] Cortese, A., 1989, DEC Challenges IBM CASE Strategy, Computerworld, 23, 41 (Oct),

[12] Carlyle, R., 1990, Is Your Data Ready for the Repository?, Datamation, 36, 1 (Jan) 43-47.

[13] Davis, A. y Sitram, P. 1994. A Concurrent Process Model for Software Development. Software Engineering Notes. Vol.19, N°2, Pág. 38-51.

[14] Fagan M.E., "Advances in Software Inspections," IEEE Transactions on Software Engineering, Vol. 12, No. 7, July, 1986, pp. 744-751.

[15] Feiler P.H. and Humphrey W.S., Software Process Development and Enactment: Concepts and Definitions, CMU/SEI-92-TR-4, ADA258465, March 1992.

[16] Fenton, N. and Pfleeger, S. Software Metrics. PWS Publishing Company. 1997.

[17] Feuche, M. Atherton, K, 1988, DEC to Boost CASE Standard, MIS Week, 9, 19 (May),33.

[18] Forte, G., 1989, In Search of the Integrated CASE Environment, C/A/S/E Outlook 89, 2.

[19] Fowler P. and Rifkin S., Software Engineering Process Group Guide, Software Engineering Institute, CMU/SEI-90-TR-24, ADA235784, September, 1990.

[20] Freedman D.P. and Weinberg G.M., Handbook of Walkthroughs, Inspections, and Technical Reviews, Third Edition, Dorset House, New York, NY, 1990.

[21] Ghezzi, C. , Jazayeri, M. , Mandrioli, D. "Fundamentals of Software Engineering". Prentice Hall. 1991.

[22] Gibson, S., 1989a, CASE Buyers Await Repository, Computerworld, 2, 20 (May), 2.

[23] Gibson, S., 1989b, Some Win, Some Lose When Repository Debuts, Computerworld, 23, 11 (Mar), 141.

[24] Graham, I. 1994, Métodos Orientados a Objetos. Adison-Wesley / Díaz de Santos,

[25] Grudin, J., 1989, The CASE Against User Interface Consistency, Comunications of the ACM, 32, 10 (Oct), 1164-74.

[26] Hecht, A. & Harris, M. 1995, A CASE Standard Interchange Format: Proposed Extension to EDIF 2.0, Cadre Technologies Inc.

[27] Humphrey W.S., "Characterizing the Software Process," IEEE Software, Vol. 5, No. 2, March, 1988, pp. 73-79.

[28] Humphrey W.S., Managing the Software Process, Addison-Wesley,Reading, MA, 1989.

[29] IEEE-STD, 1990, Computer Society Task Force on Professional Computing Tools, A standard Reference Model for Computing System Tool Interconections.

[30] IEEE-STD-610 ANSI/IEEE Std 610.12-1990, "IEEE Standard Glossary of Software Engineering Terminology," February 1991.

[31] IEEE-STD 1074-1991 Customer Request IEEE Standard for developing Software Life Cycles Process.

[32] Juristo, N. y Pazos, J. 1993. Towards a Joint Life Cycle for Software and Knowledge Engineerring.

[33] Kitson D.H. and Masters S., An Analysis of SEI Software Process Assessment Results: 1987-1991, CMU/SEI-92-TR-24, July 1992.

[34] Margolis, N., 1988, CASE Fights to Beat 'All Talk, No Action' Image, Computerworld, 22, 52 (Dec), 45-48.

[35] Martin, J.,1990, Integrated CASE Tools a Must for High-Speed Development, PC Week, 6, 3 (Jan), 78.

[36] Martin, J.; Odell, J.J. 1994, Object-Oriented Analysis and Design, Prentice Hall Inc.

[37] Myers, E., 1989, CASE Standards Connecting: Efforts Worlwide Attempting Coordination, Software Magazine, 9, 12 (Oct), 23-27.

[38] Nierstrasz, S. 1992. Component-Oriented Software Development. Communications ACM Vol.35, N°9, Pág. 160-165.

[39] Philips, B., 1989a, A CASE foe Working Together, The Electronic System Design Magazine, 19, 12 , 55-58.

[40] Paulk, M.C. Curtis, B. Chrissis, M.B., et al, Capability Maturity Model for Software, CMU/SEI-91-TR-24, ADA240603, August 1991.

[41] Paulk, M.C. Curtis, B. Chrissis, M.B. and Weber, Charles V. Capability Maturity Model for Software, Version 1.1, CMU/SEI-93-TR-24, February 1993.

[42] Paulk, M.C. Weber, C.V. Garcia, S. Chrissis, M.B. and Bush, M. Key Practices of the Capability Maturity Model, Version 1.1 (a), CMU/SEI-93-TR-25, February 1993/(Hilburn Thomas 1995).

[43] Weber, C.V. Paulk, M.C. Wise, C.J. and Withey, J.V. Key Practices of the Capability Maturity Model, CMU/SEI-91-TR-25, ADA240604, August 1991.

[44] Pfleeger, S. "Software Engineering: Theory and Practice". Prentice Hall. 1998.

[45] Phillips, B., 1989b, Software and CASE, Electronic Design, 37, 1 (Jan), 64-72.

[46] Poston, R. M., 1989, Proposed Standard Eases Tool Interconection, IEEE Software, 6, 6 (Nov), 69-71.

[47] Raccoon, P. 1994. The Chaos Model and the Chaos Life Cycle. Software Engineering Notes. Vol.20,N°1, pp 55-66

[48] Sommerville, I. "Software Engineering". Addison - Wesley. 1996.

[49] Thomas, I., 1989, PCTE Interfaces: Supporting Tools in Siftware Engineering Environments, IEEE Software, 6, 6 (Nov), 15-23.

[50] Thomas, I. & Nejmeh, B. A., 1992, Definitions of Tool Integration for Environments, IEEE Software,9,2, 29-35.

[51] Wasserman. A. I., 1988, Integration and Standarization Drive CASE Advancements, Computer Design, 27, 22.

[52] Wasserman, A., 1990, Tool Integration in Software Engineering Environments,: Proceedings of the international Workshop on Environments, F. Long (Editor), Springer-Verlag , 137-149.

[53] Yourdon, E., 1990, DEC's CASE Environment, America Programmer, 3, 1 (Jan) 4-14.

[54] Perichinsky, G., Servetto, A.C.et al. 2001. Integrated Environment of Systems Automated Engineering 2. Product Oriented. Proceedings of the XIX International Conference on Applied Informatics. (pp 706-713).Innsbruck. Austria.

[55] Perichinsky, G., Servetto, A.C.et al. 2002. Integrated Environment of Systems Automated Engineering 3. Proceedings of the XX International Conference on Applied Informatics. (accepted).Innsbruck. Austria.