

The Acceptance of Object-Oriented Development Methodologies in Australian Organizations and the Place of UML in Information Systems Programs

Richard A. Hodgett

University of South Australia, Adelaide, South Australia, Australia

Alan.hodgett@unisa.edu.au

Abstract

It is claimed that the Unified Modeling Language (UML) is emerging as the accepted standard graphical language for specifying, constructing, visualizing and documenting the object oriented information systems development process. As such it has gained a place in many information systems programs. An investigation of Australian organizations indicates that the use of object oriented development methodologies and UML is gradually increasing in Australia but is still to see general acceptance. This raises the question of the priority that should be accorded to the inclusion of UML in competition with other topics and issues in information systems education programs.

Keywords: Object-orientation, UML, methodology, systems development, education

Introduction

One of the challenges for organizations in the twenty-first century is to develop systems that ensure organizational survival and growth in a global networked economy. "There is a fundamental paradox at play in contemporary software development. On one hand, organizations are faced with the demand for faster time to market; on the other hand, these same organizations are under pressure to deliver systems with higher quality at lower cost" (Booch, 2000). Organizations are being presented with many systems analysis, systems design and systems documentation tools. "Many paradigms for systems analysis and design have been proposed over the years" (Shoval & Kabeli, 2001). "There is no shortage of information modeling methods in the field. In fact, it is a "Methodology jungle" out there" (Avison & Fitzgerald, 1996; Keng Siau, 1999).

The use of Objects and of Object Oriented Development Methodologies (OODM) has been a growing feature in specifying and developing complex information systems projects that are applying the latest information technology developments to business applications development. Prominent in this field is the Unified Modeling Language (UML). It is claimed that the UML is emerging as the accepted standard graphical language for specifying, constructing, visualizing and documenting the object oriented information systems development process. "The Unified Modeling Language (UML) has been adopted by the Object Management Group as a standard language for object-oriented systems development" (Keng Siau, 2000) and "The Unified Modeling Language, or UML (Booch, Jacobson & Rumbaugh,

1999), has rapidly emerged as a standard language and notation for object-oriented modeling in systems development," (Dobing & Parsons, 2000). While object-orientation appears to have been accepted by the information technology industry and business generally, the acceptance of total object-oriented systems development approach and the use of UML as a development tool

Material published as part of these proceedings, either on-line or in print, is copyrighted by Informing Science. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission from the publisher at Publisher@InformingScience.org

is still not universal. On one hand “Many of the language’s supporters claim that UML’s simplicity is its chief benefit (Kobryn, 1999) and argue that UML uses simple, intuitive notations that are understandable by nonprogrammers” (Keng Siau & Qing Cao, 2001) while on the other hand “Despite the increasing popularity of UML, many researchers and practitioners have raised concerns over its complexity and the lack of a comprehensive methodology to guide UML systems development process.” (Keng Siau, 2000). “In contrast to Kobryn (1999) who stated that UML is simple to use, Douglas (1998) argued that UML is somewhat large and complex, which can be daunting to novice users. He called for future research to examine the questions: (i) How easy is it to understand UML? And (ii) How easy is it to use UML?” (Keng Siau & Qing Cao, 2001). There are grounds to believe that industry might be happier to stay with the proven Entity-Relationship (E-R) model and that it may be considered that UML is deficient in capturing certain data concepts and business rules. “Some empirical studies indicate that Entity Relationship (ER) schemas are often more correct and easier to develop than corresponding OO schemas.” (Shoval & Shiran, 1997; Haplin & Bloesch, 1999) and “UML’s object-oriented approach facilitates the transition to object-oriented code. As shown later however, UML can make it awkward to capture and validate data concepts and business rules with domain experts, and cater for structural changes in the application. These problems can be remedied by using a fact-oriented approach where communications takes place in simple sentences, each sentence type can easily be populated with multiple instances, and attributes are eschewed in the basic model”(Haplin & Bloesch, 1999). In the light of this debate or difference in views it is of interest to explore industry’s views and experience with Object-oriented Development Methodologies and UML. This paper reports on the results of a study into the use of UML and object oriented methodologies in specifying complex industry information systems projects that are applying the latest information technology developments to business applications in Australia to meet the challenges of a global economy.

Methodology

A survey form was mailed to information systems managers, or executives in similar positions, in a sample of businesses across Australia. The survey requested information on the use of object-oriented development methodologies, UML and other object-oriented tools. The response rate was low, 12 percent of questionnaires mailed in fact, which was an interesting result in itself. Falconer & Hodgett (1999) have explored reasons for low response rates to surveys. Because of this low result a series of interviews were conducted with managers in a variety of organizations. In interpreting the results it should be borne in mind that Australian businesses generally are smaller and employ fewer people than those in countries which have been the subject of much research; for example, Lederer and Mendelow (1988) surveyed organizations with up to 125,000 employees and Earl (1993) surveyed organizations having average annual revenues of £4.5 billion. Even the largest Australian organizations are small by comparison to these. The Australian Bureau of Statistics (2001) reports on the Australian business sector in six employment size groupings, the largest being those businesses employing more than 200 people. In 1999-2000 there were 33,200 organizations with 20 - 99 employees, 3700 with 100 - 199 employees and 2700 with more than 200 employees. This paper reports on both the results of the survey and the outcomes of the interviews

Results

Of those companies surveyed and interviewed only some seventeen percent were using object-oriented methodologies in their systems development while only some thirteen percent have ever tried to use UML. The interviews provided some insight into why this should be so. The interviews suggested that in part the lack of take up was because of organizational size and take-up cost. As shown above Australian businesses generally are smaller and employ fewer people than those in countries that have been the subject of much research. It was explained by managers that changing development methodologies would involve high upfront costs for most companies and that in many instances these costs would have to be

written off. Evidently the smaller the organization the less the necessity to write off expenditure would appeal, particularly if it was felt that the existing situation was satisfactory. It was felt that many managers did not want to leave what was described as a comfort zone, particularly in view of the empirical studies indicate that Entity Relationship (ER) schemas are often more correct and easier to develop than corresponding OO schemas. (Shoval & Shiran, 1997; Haplin & Bloesch, 1999). It was also suggested that a spill over from this comfort zone was that many of the technical leaders in the organizations have not kept up-to-date in the new methodologies and were therefore not comfortable with introducing them or using them in their organizations. The change to OODM is a mindset problem for people used to ER and older methodologies. This in turn seems to give rise to a "Catch 22" situation in relation to tertiary education. Industry, through the attitudes of a number of managers, is not interested in recruiting staff with knowledge of OODMs. Because of this the tertiary institutions do not place an emphasis on OODMs as part of the skill set of graduates thus reinforcing the lack of knowledge and interest in these methodologies on the part of new entrants into the organizations.

The responses and comments of those that use or had used UML showed that few used or continued to use all the views or aspects of the modeling language. The parts selected depended on the application and the comments indicated that part of the lack of support was because most business applications are not real time and do not have the degree of size or complexity that might justify the use of a wider range of UML views. It was recognized that Use Cases were the core of the methodology. Class and Object diagrams followed as the most accepted parts of the language while Activity diagrams were universally disregarded. Criticism of these latter diagrams seemed to center around the time taken to model minutia while comments from programmers indicated that this level of detail was not required.

Questions relating to the use of a number of specific object-oriented modeling tools or packages, whether using UML or not, indicated that even among the low percentage still using object-oriented methodologies no preference was shown for any one tool; indeed in a number of instances the continuing use of object-orientation seemed to be confined to object-oriented programming languages only.

A further issue was raised in this investigation that related to the use and understanding of object oriented models and programming languages. It was suggested that because many of the technical leaders in the organizations have not kept up-to-date in the new methodologies that there was a misunderstanding about their role and use. It appears that in a number of instances the orientation of OODMs and the use of UML toward modeling business processes has not been recognized and they are being used in an environment that sees them as tools in a data-modeling environment. In such situations the true potential of these tools is not being realized with object oriented programming languages being used extensively for procedural programming and objects being developed with a proliferation of methods that ultimately defeats the whole concept of reusability.

Conclusions

The use of OODMS and UML is gradually increasing in Australia but is still to see general acceptance. A number of factors can be suggested for the reticence in adopting the new development methodologies. There seems to be a sense that the existing methodologies in use are doing an adequate job in an environment characterized by the relative lack of size of businesses, comparative lack of complexity of systems and the small demand for twenty-four hour seven days per week on-line real-time systems. In view of this there is a reluctance to divert the time and resources needed to make such fundamental changes. The up front costs required and the possibility that they may be written off is a disincentive. This is exacerbated by a lack of understanding about the true nature and potential benefits of OODMs. This, then, raises the question of the priority that should be accorded to the inclusion of UML in information systems education programs. It would seem that other topics and issues should be given a higher priority.

References

- Avison, D.E., & Fitzgerald, G. (1995). *Information systems development: Methodologies, techniques, and tools*, 2nd Edition, McGraw-Hill, London.
- Booch G. (2000). Unifying enterprise development teams with the UML. *Journal of Database Management*, Vol. 10 No. 4, Oct-Dec.
- Booch, G., Jacobson I., & Rumbaugh, J. (1999). *The unified modeling language user guide*. Reading, MA: Addison-Wesley.
- Dobing B. & Parsons J. (2000, Oct-Dec). Understanding the role of use cases in UML: A review and research agenda. *Journal of Database Management*.
- Douglass, B.P. (1998). UML for Systems Engineering. *Computer Design's: Electronic Systems Technology & Design*, (37:11).
- Earl, M.J. (1993, March). Experiences in strategic information systems planning. *MIS Quarterly*.
- Falconer, D.J., & Hodgett, R.A. (1999). Why executives don't respond to your survey. *Proceedings of the Australasian Conference on Information Systems*, Wellington, NZ.
- Haplin T. & Bloesch A. (1999, Oct-Dec). Data modeling in UML and ORM: A comparison. *Journal of Database Management*, Vol. 10 No. 4.
- Keng Siau & Qing Cao. (2001, Jan-Mar). Unified modeling language (UML) – A complexity analysis. *Journal of Database Management*.
- Keng Siau. (1999, Oct-Dec). Information modeling and method engineering: A psychological perspective. *Journal of Database Management*, Vol. 10 No. 4.
- Keng Siau. (2000, Oct-Dec). Systems analysis and design using UML. *Journal of Database Management*.
- Kobryn, C. (1999). UML for Systems Engineering. *Communications of the ACM* (42:10) pp29-38
- Lederer, A.L. & Mendelow, A.L. (1988). Convincing top management of the strategic potential of information systems. *MIS Quarterly*, Dec, 525–534.
- Shoval P. & Kabeli J. (2001, Jan-Mar). FOOM: Functional and object-oriented analysis & design of information systems: An integrated methodology. *Journal of Database Management*.
- Shoval P. & Shiran, S. (1997). Entity-relationship and object-oriented data modeling – an experimental comparison of design quality: *Data & Knowledge Engineering*, 21(3).
- Small Business in Australia Update 1999-2000. (2001). Australian Bureau of Statistics.

Biography

Richard Alan Hodgett, MCA, BCA, BA, is a Senior Lecturer at the University of South Australia, Australia. He spent the first part of his working career as a telecommunications engineer with the Automatic Telephone and Electric Company, England. He moved to New Zealand in the mid 1950's and worked with the New Zealand Post Office in the engineering branch. In the late 1960s, he transferred to the Electronic Data Processing Division as a programmer and then analyst. He joined Victoria University of Wellington, New Zealand in 1974, and then moved to Australia in 1989 and joined the University of South Australia.

His current research interests lie with data management in organizations, the integration of process control and organizational information systems and the development of effective information systems.