

Object Oriented Analysis and Design: Do We Need More UML in the Classroom?

Richard V. McCarthy

Bruce White

Computer Information Systems, Quinnipiac University

Hamden, CT 06518, USA

Richard.mccarthy@quinnipiac.edu

Bruce.white@quinnipiac.edu

Martin Grossman

Department of Management, Bridgewater State College,

Bridgewater, MA 02325

mgrossman@bridgew.edu

Abstract

UML has emerged as the de facto standard for object oriented analysis and design. It is a complex notational and symbolic language with many features and functions that is methodology independent. A qualitative and quantitative survey of UML users was conducted to determine the extent to which UML meets their needs. This research evaluates the qualitative responses to provide a basis to examine; *to what extent do we need to include UML within IT curriculum?*

Keywords: UML, object-oriented analysis and design

1. INTRODUCTION

The Unified Modeling Language (UML) was introduced over 7 years ago as a means to support the design and documentation of objects. UML is a visual language that utilizes numerous diagrams and notations to express objects to define them in discrete terms. It is a general purpose language that can be used with all object methods within any application domain; as such it is both hardware and software independent.

With the advent of more mainstream object oriented programming environments (i.e., .NET and Java) UML is being used to a much greater extent than had been in the past. This research reports on the results of a qualitative study in the use of UML. Respondents were asked to comment on their use and experience with

UML. Based upon these results we conclude with a discussion of the future of UML in the curriculum of undergraduate information technology programs.

2. UNIFIED MODELING LANGUAGE

UML is a complex tool that although most commonly associated with object-oriented technologies, can be used to model any type of application in any type of environment. Its focus has been on design rather than execution, however many UML tools support a wide array of code generation, reverse engineering and testing functions. UML can be used to express the results of analysis and design from any methodology. UML (version 2.0) defines 12 types of diagrams that are divided into three categories (static application structure, dynamic behavior, application module management). The

Object Management Group (www.omg.org) defines three steps for starting a UML project:

1. Select a methodology – to define the process for gathering requirements
2. Select a UML development tool
3. Get training – its best to get training on the particular tools within UML that will be utilized as UML is a large and complex process.

3. UML IN THE CURRICULUM

Johnson (2000) identified three components of UML that could be emphasized early into an information technology curriculum:

1. Class diagrams – which are used to describe static relationships between classes in terms of attributes and methods.
2. Use Case diagrams – which are used to help define the scope of a system by providing an explanation of the user's view of the problem.
3. Activity diagrams – used to identify decision points.

Satzinger and Jackson (2004) point out that although object-oriented analysis and programming seems well understood by industry and educators, there are two gaps in understanding:

1. The processes, techniques and artifacts required to bridge the gap between object-oriented analysis and object-oriented programming.
2. An understating of the development process for building an object-oriented system.

They recommend a structured curriculum that integrates UML models to depict requirements for object oriented programming classes to provide students with an iterative approach to learning UML (Jackson and Satzinger, 2003).

LeBlanc and Stiller (2000) concluded that although it is nearly impossible to present all of the details of each facet of UML in a single semester course, UML can be effectively used to provide students with a means to describe a model of a system

that is unambiguous, concise and supports various levels of abstraction.

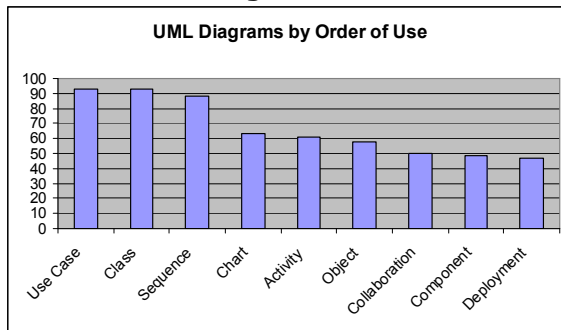
However, in a study of UML complexity, Erickson and Siau (2004) found that four diagrams (Class, Use Case, Sequence and Statechart) clearly were distinguished as being more important and provided a practical starting point for UML training.

4. METHODOLOGY

A quantitative and qualitative survey of UML user was conducted using a web based survey. The quantitative results were designed to test the task-technology fit of UML and the results were inconclusive (Grossman, McCarthy and Aronson, 2004). The survey was distributed to 1,507 UML users who were members of object oriented analysis and design online user groups (e.g. The UML Forum, UML Café, Objects by Design Forum, UML Zone, The Precise UML newsgroup, Rational Unified Process Forum, Sparx System Forum, Rose Forum, Object Technology User Group). The request to participate in the survey was sent via email, with a link to the web survey. These forums were used specifically to identify participants who were experienced in the use of UML. Of the 1,507 e-mails initially sent, 133 did not reach their intended recipient, and bounced back. Of the remaining 1,374 emails, a total of 150 users responded to the survey (over 83% of whom responded within 72 hours from the time the emails were initially sent.). This represented a response rate of 10.91%.

The respondents were asked to indicate which of the UML diagrams they used (see Figure 1), from a list provided within the survey. The list contained the 9 diagrams that are part of the UML version 1.0 standard. UML version 2.0 now contains 12 diagrams; however, the standard is still new and has not yet been widely adopted.

Figure 1 –Percentage of use UML Diagrams



Although the Use Case, Class and Sequence diagrams indicated a greater percentage of usage, the responses indicate that all 9 of the UML diagrams are used in varying, but important degrees.

5. QUALITATIVE RESULTS

Thirty-seven respondents commented upon their experience with UML. The responses included both the individuals' personal experience with UML and how they perceived its use. Some of the respondents were enthusiastic in their response. Including one individual who commented, "We have made UML the core of our design and development process. We are committed to UML because we believe it enables us to produce the products within the timescales". Others were far more guarded in their opinion of the effectiveness of UML. One respondent commented, "UML is not a complete and comprehensive solution to all of the challenges associated with defining requirements and designing software systems. It is however, a useful and valuable technique. It is by no means the only technique that an analyst or designer needs any more than a hammer is the only tool that a carpenter needs..."

6. UML AS A COMMUNICATION TOOL

Several of the respondents commented on the use of UML as a communication tool. One of the respondents indicated, "Many of the questions imply that UML is more of a method than a way of communicating -- and to many, I think it is. But to me, it is a way of communicating, and my only use for it is to allow the team to guess enough

about the solution to begin writing tests and code. I use UML to communicate an idea that is too complex to describe by hand-waving. The moment we understand each other enough to start writing code, we stop writing UML. And the moment the code teaches us more than the UML diagram did, we abandon (rather than update) the diagram. We are also very sloppy with notation as long as the diagram communicates what we need to say. For example, we might start with an object diagram and then mix a piece of object diagram in. Our use of UML is therefore somewhat perfunctory compared to what the questions in the survey lead me to think might be the norm (or, in any case, might be the desired sample group for the survey). If the way I use UML is not what you mean when you say UML, you might want to consider me an outlier."

While another respondent included, "UML is an industry standard and should be used in all projects, either object oriented or not. Communication in team can be improved considerably with UML. Don't forget: you must have an appropriate process to be able to use UML correctly." Another respondent included, "UML is very good for modeling reactive systems."

7. UML TRAINING

Training was the issue that received the most significant attention amongst the respondents. Those who commented on training pointed out two important points must be addressed; the need to understand the tool and its capabilities, and the need to understand how to use UML as a communication tool to improving the understanding of system requirements. One respondent commented, "UML is quite useful but adequate training is quite lacking. To the point UML training is about training people on how to think about problems and devise efficient solutions quickly. The notation itself is just the medium. Also, over-engineered systems are (wrongly) linked to UML and this leads to a bad perception of it. UML can be simple and efficient, thought provoking and on the contrary help in *simplifying* overly complex designs or concepts. In that UML can add significant shareholder value. In

being wrongly used it can for sure be subtracting value."

Understanding UML requires training and experience, as its usage continues to grow it will be increasingly important to ensure that users understand its capabilities and its limitations. Its increasing importance has been recognized within the IS Model Curriculum as it is specifically identified within the IS 2002.7 Analysis and Logical Design course and the IS2002.8 Physical Design and Implementation with DBMS course.

8. DISCUSSION

UML continues to expand in use throughout the industry in part because object-oriented technologies such as VB .NET and Java have continued to increase in popularity and in part because UML is technology independent. "We have just started using UML for our projects. Most of the code will be written in Mainframe COBOL, so we won't have much use for the OO Code Generation found in most UML based tools. So far, I see the benefits of using UML to be in the communication with our user departments (various diagrams) and estimating the projects (Use Cases).", indicated one respondent. UML has emerged as the standard for object-oriented analysis and design, and it has been demonstrated that it can be used as an effective communication tool that is methodology independent.

The IS2002 model curriculum has recognized the growing importance of UML and has recommended that it be included in the learning objectives of database management and analysis and design courses. In the discussion section on the IS2002.7 course, it is noted that "Students will be exposed to methods to support each stage of the development process." The IS2002.8 Physical Design and Implementation with DBMS course indicates the emphasis on tools like UML is stronger. In the topics sub-section the model curriculum reads: "Conceptual, logical and physical data models and modeling tools; structured and *object design* approaches; models for databases; relational and object oriented; design

tools, data dictionaries; repositories ..." (Gorgone, et al, 2002)

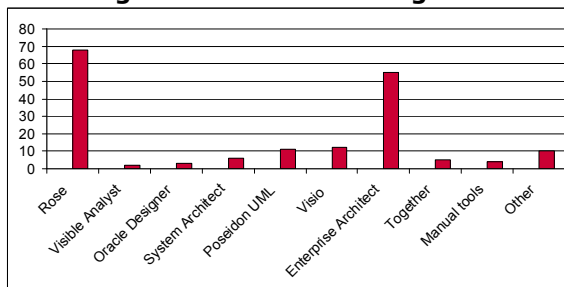
At this point in time, comparatively few universities or colleges have fully subscribed to the model curriculum or have achieved IS accreditation. *To what extent is our curriculum deficient in this area?* A future study is planned to survey the depth of UML coverage in existing IS curriculum and the planned curriculum changes to increase its coverage.

UML is a complex and powerful tool; it must be carefully integrated throughout an IS curriculum to take advantage of its abilities. One survey respondent commented, "Because of the breadth and depth of the UML construct, it is unlikely, indeed unreasonable, to expect or mandate the use of the UML in its' entirety for any given project. Rather, like any true framework, it should be used discriminately to aid in the consistent analysis, documentation and communication of particularly challenging elements of the system design and development. The only possible exception is in gathering and documenting system and user requirements. It is our belief that successful OOA/OOD is accomplished when the solution reality accurately reflects the requirements rather than the real-world reality. As a result, we find it useful to fully employ the requirement notation and selectively employ most other diagrams with a fairly consistent focus on activity, sequence, collaboration and class diagrams. This provides a workable balance between the challenges of effective project (cost) management and mitigating risks associated with poor design." It has become increasingly important to expand the depth and breadth of understanding of UML; however we are not yet at the point where it will be the only methodology in use. This poses the question; *How much of UML can we integrate into IS curriculum?* The answer to this question is dynamic, not static. There are many methodologies, *so is it important that a student thoroughly understand each methodology?* Or, is it more important that students understand the value of methodologies, how and when to apply them, where to obtain more information when needed, and how to define, evaluate and design effective and

efficient information systems. We purport that the answer to this question is to design curriculum that iteratively transitions towards an increased emphasis on understanding UML and its important role in the design and development of software applications.

There are many products available that assist in the creation of UML diagrams; these have varying degrees of complexity and functionality. Figure 2 displays the distribution of UML tool usage by the survey respondents. It is interesting to note that one of the most popular products is Rational Rose which was recently acquired by IBM Corporation. Visible Analyst, which was used by very few of our respondents has been available at a nominal cost with several popular systems analysis & design and database textbooks for several years. Rational Rose, which in the past was not available for academic licensing, is now available free of charge through the IBM university network program. Additionally, Poseidon UML offers its *Community Edition 2.5* as a free download from www.gentleware.com. It will be interesting to see what impact this has in the future.

Figure 2 - UML Tool Usage



9. REFERENCES

- Ericson, J. and Siau, K, (2004), "Theoretical and practical complexity of UML", *proceedings of the Tenth Americas Conference on Information Systems*, New York, NY, August 6-8, 2004, 1669-1674.
- Gorgone, John T., Davis, Gordon B., Valacich, Joseph S., Topi, Heikki, Feinstein, David L., and Longenecker, Herbert E. Jr. (2002), "IS 2002 model curriculum and guidelines for

undergraduate degree programs in information systems"

- Grossman, M., McCarthy, R.V., and Aronson, J.E., Does UML Make the Grade? Insights from the Software Development Community, *Information & Software Technology*, (accepted, September 2004).
- Jackson, R.B., Satzinger, J. W., (2003) "Teaching the complete object-oriented development cycle, with OOA and OOD, with UML and the UP., *Information Systems Education Journal (ISEDJ)*, 2003, 3-20.
- Johnson, D. (2000), "UML basics – Where do they fir in the classroom?" *proceedings of the Consortium for Computing in Small Colleges Midwestern Conference*, 94-95.
- LeBlanc, C. and Stiller, E., (2000, May), "UML for undergraduate software engineering", *proceedings of the Consortium for Computing in Small Colleges Northeastern Conference*, 8-18.
- Satzinger, John W., Jackson, Robert B., (2003) "Making the Transition from OO Analysis to OO Design with the Unified Process, *proceedings of the Ninth Americas Conference on Information Systems (AMCIS)* August 4-6, 2003, 3200-3210.