

Many UML projects are not Use Case driven.

How UML IS USED

The Unified Modeling Language (UML) emerged in the mid-1990s through the combination of previously competing object-oriented (OO) software engineering methods developed by Booch [1], Jacobson et al. [6], Rumbaugh et al. [8], and others. Control over its formal evolution was placed in the hands of the Object Management Group (www.omg.org), and the language has become widely accepted as a modeling standard for OO software development. A large number of practitioner articles and books, and some contributions by academic researchers, have been devoted to articulating various aspects of the language, including guidelines for using it [5]. UML per se is a language, not a methodology, so it is not surprising these guidelines are not always consistent [4].

Despite widespread interest in UML, there is little quantitative evidence on the level and nature of UML use. Our study examines seven

UML components used in systems analysis and design and addresses three key questions. First, to what extent are these UML analysis components being used and for what purposes? Second, do differences in the levels of component use and the reasons for these differences reflect the apparent complexity of the language [7, 9]? Third, how successful is UML in facilitating communication within software development teams?

The research began with a review of the UML literature and some preliminary interviews with UML practitioners and their clients. Following this, we developed a Web survey targeted at analysts familiar with OO techniques and UML in particular. UML 1.5 contains nine basic diagrams [3]—six of these (Use Case, Class, Activity, Collaboration, Sequence, and Statechart Diagrams) plus Use Case Narratives were determined to be most closely related to the research questions. The Object Diagram, which is closely related to the Class Diagram, and the Component and Deployment Diagrams, used in application architecture modeling, were excluded as less relevant and to keep the survey to a reasonable

BY BRIAN DOBING AND JEFFREY PARSONS

length. The OMG agreed to support the project and informed their members of the survey. Those contacted directly were encouraged to share the link with other UML users in their organizations. A link to the survey was also provided from the main OMG Web page, and some others learned about it through other sources, so not all respondents were from OMG member organizations. No participation incentive was offered. This article is based on data collected from March 2003 to March 2004 and thus most responses reflect experiences with UML 1.5. Both the survey and this article use UML 1.5 terminology (for example, “Collaboration Diagrams” rather than the newer “Communication Diagrams”).

RESULTS

The Web survey attracted 171 usable responses from analysts using UML and another 11 from those using UML components as part of another OO methodology. Respondents reported having been involved in an average of 27 projects (about 6.2 using UML), over an average 15-year career (4.7 using UML) in information technology. The median “typical” UML project had a budget of around \$1,000,000 and 6.5 person-years, and required about 50,000 lines of code.

For each of the major UML analysis components, Figure 1 shows the percentage of respondents who reported it being used in at least two-thirds of their projects, the percentage who never used it, and the percentage of users who believed it provided new information not found in Use Case Narratives. Although UML is often presented as Use Case driven [5], Class Diagrams were the most frequently used component, with 73% of respondents saying they were used in two-thirds or more of their projects. Use Case Narratives (44%) were ranked fourth. However, respondents were generally familiar with all the components, ranging from only 3% who reported Class Diagrams were never used in their

projects to 25% for Collaboration Diagrams. Those with the most UML experience reported their projects used more components, suggesting that usage levels might increase as practitioners gain experience.

Taking a Use Case driven perspective, the survey asked which components provide new information beyond that contained in Use Case Narratives. The question—asked only of those respondents whose projects used both Use Case Narratives and the UML component in question—used a five-point scale from “No New Info” to “All New Info,” with “Some New Info” as the midpoint. Figure 1 shows the percentage of respondents who chose the midpoint or higher on this scale.

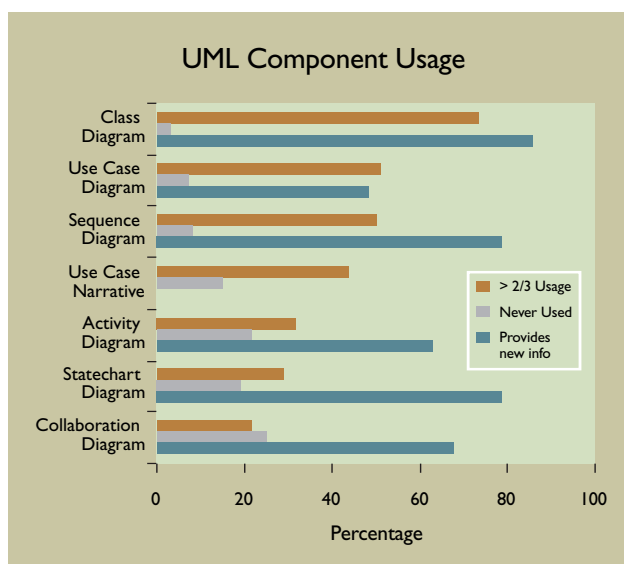


Figure 1. Use and perceived information added of UML components.

Usage rates are not well explained by how much new information is provided. While the Class Diagram was rated highest for both usage and providing new information, Statechart Diagrams were used much less frequently, but ranked second in providing new information. Sequence Diagrams were rated as slightly less useful in providing new information, but are used more frequently. The Use Case Diagram was rated as least useful in providing additional

information, consistent with its role of presenting an overview of the project. This might indicate that providing the same information in a different form can also be valuable.

Survey respondents do not appear to trade off between diagrams. For example, given that Sequence Diagrams and Collaboration Diagrams are “isomorphic” [3], one might expect projects to use either the Collaboration Diagram or the Sequence Diagram but not both. However, those who used one were more likely to use the other and, in general, usage rates of the different UML components were all positively correlated.

Those who reported using a particular component in at least one-third of their projects were asked to rate its usefulness (on a five-point scale from “Not Useful” to “Essential”) for four possible purposes. The number of respondents ranged from 54 for Collaboration Diagrams to 115 for Class Diagrams. Table 1 reports the percentage who rated each component as “useful,” ranging from “Moderately Useful” (midpoint on the

scale) to “Essential” for each purpose. For example, 87% of respondents rated Use Case Narratives as useful for “verifying and validating requirements with client representatives on the project team.” The use of other components for this purpose was higher than expected, based on literature that strongly emphasizes Use Cases when working with clients. In contrast, the Class and Sequence Diagrams were most useful for “clarifying understanding of application among technical members of the project team.”

As noted by Dobing and Parsons [4], a potential disconnect could result from relying on Use Case Narratives when working with clients and Class Diagrams when working with technical team members. However, all components received at least “moderately useful” ratings from half or more of the respondents across all forms of communication. This suggests the disconnect problem might well have been addressed in practice, if not in the UML literature.

Table 2 shows why infrequent (less than 1/3 of the time) or non-users were not using components more often. The number of respondents was smaller, ranging from only eight for Class Diagrams to 59 for Collaboration Diagrams. They could select more than one reason for each. A lack of understanding by analysts is the primary factor among the few not using Class Diagrams (50%). Similar concerns were expressed by 48% of respondents about Activity Diagrams. Leading concerns for the remaining components are about usefulness (Statechart), value (Sequence and Use Case Diagrams and Narratives), and redundancy (Collaboration). Statechart Diagrams seem to be less useful most of the time (Table 2) but are rated highly for providing new information in some situations (Figure 1) and have low redundancy (Table 2). As one

UML Component	Client Verification	Clarifying Tech Understanding	Programmer Specifications	Maintenance Documentation
Use Case Narrative	87	74	79	68
Activity Diagram	77	80	81	73
Use Case Diagram	74	66	62	61
Class Diagram	57	93	89	92
Sequence Diagram	62	91	84	85
Statechart Diagram	50	82	79	71
Collaboration Diagram	50	74	70	62

Table 1. Percentage using each UML component for key purposes.

interview subject put it, “When they are useful, they are very useful.”

The differences between Collaboration and Sequence Diagrams are striking, given that the diagrams are isomorphic. Respondents in our survey clearly prefer Sequence Diagrams to Collaboration Diagrams. The latter are used less frequently (Figure 1), are less useful for various purposes (Table 1), and are more likely to be seen as redundant (Table 2) when compared to the former.

User participation has long been considered crucial

to the system development process. The survey also asked about the client’s role in relation to each of the UML components being studied. Respondents were able to select more than one (for example, they could help to develop Use Case Narratives, review some or all of them upon completion, and have formal approval authority). The results are summarized in Figure 2. For example, 76% of respondents who used Use Case Narratives and answered this question report that clients are

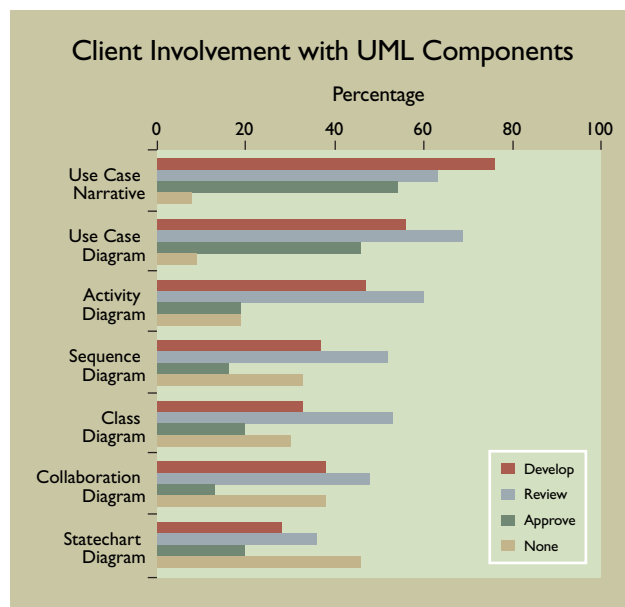


Figure 2. Client involvement.

involved in Use Case development at some level. The “None” column shows the number explicitly indicating no client involvement.

The results show that clients are most likely to be involved in developing, reviewing, and approving the Use Case Narratives and associated Use Case Diagram. However, client involvement levels across the remaining components during development seem higher than would be expected from the literature. Formal client approval, on the other hand, is generally low. When asked whether the UML facilitated communication with clients, 55% said it was at best “Moderately Successful.”

W

hile the high level of client involvement across all UML analysis components may address earlier concerns about the potential communication disconnect, these involvement levels may not be typical of UML projects as use of the language increases. The survey did not ask about client characteristics but, based on the preliminary interviews, early UML clients tend to have the characteristics attributed to early adopters in the innovation literature. These include higher levels of education, prior experience with IT, and an interest in experimenting with new ideas and technology. Some attended the same UML training sessions as the IS members of the team. All interviewed clients welcomed the use of the UML and some showed considerable insight, even questioning whether the analysts were using it appropriately.

Respondents were asked how many years of experience they had with OO programming, OO analysis and design, and UML. Respondents were divided into three similar-sized groups (less, average, and greater experience) based on their responses to these items. Generally, those with more UML experience made more use of UML components and for more purposes than those with less experience. This suggests that analysts need time to learn how to use UML well. Some survey respondents had not used UML and were instead asked why not. The majority indicated “too few people familiar with the UML.” Thus, ongoing training programs are critical.

CONCLUSION

This survey is the first we are aware of investigating how and why UML analysis components are used. Overall component use was similar to an earlier study that found highest usage levels for Use Case Diagrams and Class Diagrams and lowest for Collaboration Diagrams [10]. But reported levels of regular usage of UML components were lower than expected. Many projects are not Use Case driven, with only about half of respondents using them regularly (two-thirds of the time or more). Only Class Diagrams are being used regularly by over half the

respondents, with Sequence and Use Case Diagrams used by about half. “Not well understood by analysts” was the most frequent by those not using a particular UML component. The second most frequent explanation was “insufficient value to justify the cost.” This may well be true for some projects, but perhaps many analysts do not yet know how to use the components to their full advantage.

These explanations support the argument that the UML may be too complex. Perhaps the first UML projects undertaken by organizations and analysts should avoid Collaboration Diagrams which, based

UML Component	Not well understood by analysts	Not useful for most projects	Insufficient value to justify cost	Information captured would be redundant	Not useful with clients	Not useful with programmers
Class Diagram	50	13	13	25	25	25
Sequence	32	23	36	14	23	23
Use Case Narrative	29	26	37	29	11	26
Use Case Diagram	32	32	42	19	29	42
Statechart	35	42	28	12	28	33
Activity	48	23	35	35	14	25
Collaboration	27	32	24	49	29	24

Table 2. Reasons for not using some UML components (% responses).

on our findings, are used much less often, deemed to be less useful, and appear to offer little additional value in relation to Sequence Diagrams. Statechart Diagrams are very useful for their intended purpose but are not critical for many systems. Focusing on a smaller set of components, as proposed by Ambler [1], may be a better strategy for both analysts and students in the early stages of learning UML, and may reduce the cost of ensuring consistency across different components.

The results also suggest that more extensive educational programs are needed, both to increase the number of analysts familiar with UML and provide ongoing support to help them make fuller use of its capabilities. In addition, more attention may be needed on the issue of how clients/users can be better prepared to participate in development and review of artifacts beyond Use Case Narratives. Respondents reported relatively high levels of formal client involvement in producing, reviewing, and approving each UML component. This suggests that UML should not be considered exclusively as a language for software professionals, and that a greater understanding of UML diagrams and their roles in building systems is needed throughout organizations. Standardization of UML has made a major contribution toward this goal; standardization of usage guidelines is needed as well.

SUMMARY

- The frequency of use of UML components varies considerably: Class, Sequence, and Use Case Diagrams are used most often, while Collaboration Diagrams are used least.
- Apparently, at least half of UML projects are not Use Case driven: Class Diagram use substantially exceeds Use Case Diagram and Narrative use.
- Contrary to claims in the popular literature, developers appear to believe that UML diagrams can be understood by clients: Clients are most involved with Use Case Narratives and Activity Diagrams, but are more involved with the remaining components than we expected.
- While systems analysts and programmers rely most on Class and Sequence Diagrams they also use the Use Case Narratives, suggesting that the potential communication disconnect may not be a concern in practice.
- Use Case Narratives appear not to capture all requirements: Class, Sequence, and Statechart Diagrams provide the most additional information beyond Use Case Narratives.
- The complexity of UML is a concern, suggesting more programs are needed to help IS professionals and their clients learn the language and how to use it more effectively. ■

REFERENCES

1. Ambler, S. *Agile Modeling: Effective Practices for Extreme Programming and Unified Process*. John Wiley, New York, 2002.
2. Booch, G. *Object-Oriented Analysis and Design with Applications, 2nd ed.* Benjamin/Cummings, Redwood City, CA, 1994.
3. Booch, G., Rumbaugh, J., and Jacobson, I. *The Unified Modeling Language User Guide*. Addison-Wesley, Reading, MA, 1999.
4. Dobing, B. and Parsons, J. Understanding the role of Use Cases in UML: A review and research agenda. *Journal of Database Management* 11, 4 (2000), 28–36.
5. Jacobson, I., Booch, G., and Rumbaugh, J. *The Unified Software Development Process*. Addison-Wesley, Reading, MA, 1999.
6. Jacobson, I., Christerson, M., Jonsson, P., and Overgaard, G. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, Reading, MA, 1992.
7. Kobryn, C. Will UML 2.0 Be agile or awkward? *Commun. ACM* 45, 1 (2002), 107–110.
8. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorenzen, W. *Object-Oriented Modeling and Design*. Prentice Hall, 1991.
9. Siau, K. and Cao, Q. How complex is the Unified Modeling Language? *Advanced Topics in Database Research 1*. Idea Publishing Group, Hershey, PA, 2002, 294–306.
10. Zeichick, A. Modeling usage low; Developers confused about UML 2.0, MDA. *SD Times*, July 15, 2002. (Available at www.sdtimes.com/news/058/story3.htm).

BRIAN DOBING (brian.dobing@uleth.ca) is an associate professor of information systems in the Faculty of Management, University of Lethbridge, Lethbridge, Alberta, Canada.

JEFFREY PARSONS (jeffreyp@mun.ca) is a professor of information systems and associate dean (research) in the Faculty of Business Administration, Memorial University of Newfoundland, St. John's, Newfoundland and Labrador, Canada.

UML should not be considered exclusively as a language for software professionals; a greater understanding of UML diagrams and their roles in building systems is needed throughout organizations. Standardization of UML has made a major contribution toward this goal; standardization of usage guidelines is needed as well.

