The Unified Modeling Language: An inquiry into current practices and user perceptions

Martin Grossman
Nova Southeastern University

Richard McCarthy
Quinnipiac University

Jay Aronson
University of Georgia

Follow this and additional works at: http://aisel.aisnet.org/amcis2004

Recommended Citation
http://aisel.aisnet.org/amcis2004/205

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2004 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
The Unified Modeling Language:  
An inquiry into current practices and user perceptions

Martin Grossman
Nova Southeastern University
grossma@nova.edu

Richard V. McCarthy
Quinnipiac University
Richard.McCarthy@quinnipiac.edu

Jay E. Aronson
University of Georgia
jaronson@uga.edu

ABSTRACT
The Unified Modeling Language (UML) has become the de facto standard for systems development and has been promoted as a technology that will help solve some of the longstanding problems in the software industry. However, there is still little empirical evidence supporting the claim that UML is an effective approach to modeling software systems. Indeed, there is much anecdotal evidence suggesting the contrary, i.e. that UML is overly complex, inconsistent, incomplete and difficult to learn. This paper describes an investigation into the adoption and use of UML in the software development community. A web-based survey was conducted eliciting responses from users of UML throughout the world. Results indicate a wide diversity of opinion regarding UML, reflecting the relative immaturity of the technology as well as the controversy over its effectiveness. This paper discusses the results of the survey and charts the course for future research in UML usage.

Keywords
Unified Modeling Language, UML, object-oriented analysis and design, OOAD, task-technology fit

INTRODUCTION
Object-oriented technology has profoundly changed the way software systems are designed and developed (Yourdon, 1996). OO proponents are quick to claim the advantages over the traditional structured or process oriented (PO) approaches, such as easier modeling, increased code reuse, higher system quality, and easier maintenance (Garceau, Jancura and Kneiss, 1993; Johnson, 2000). Indeed, object-oriented technology has often been promoted as a silver bullet, capable of solving many of the longstanding ills facing the software industry.

The advent of the Unified Modeling Language (UML) has fueled the continued growth and acceptance of object-oriented technology. UML is a visual modeling language, composed of notations and textual components to express object-oriented system designs (Fowler and Scott, 2004). During the early 90s, the object-oriented methods landscape was one of contention and confusion. Prior to 1994, there were many competing visual modeling languages and methodologies on the market. All of these had their loyal followers as well as their detractors, and the selection of one technique over another was not an easy choice. The impetus behind UML was to fuse, or unify, the best practices from the strongest of these methods. Ultimately, three methods emerged as the primary contenders: the Booch Method (Booch, 1994), the Object Modeling Technique or OMT (Rumbaugh Blaha, Premerlani, Eddy and Lorenson, 1991), and the Objectory Method (Jacobson, Jonsson and Overgaard, 1992). Elements from each of these methods make up the core of UML, and the primary authors, better known as ‘the Three Amigos,’ are still working on the ever evolving UML specification, along with many other participants, under the tutelage of the Object Management Group (OMG).

Although UML has achieved tremendous popularity and is rapidly becoming the standard for object-oriented systems development, there are many who feel that it is too difficult to use and that it is not fulfilling its promise. Commonly heard complaints about UML are that it is too big and complex, it is semantically imprecise, it is implemented in a non-standard manner, it has limited customizability, it has inadequate support for component based development, and that it is unable to
easily interchange model diagrams (Kobryn, 2002). Much of the existing literature relating to UML usage focuses on such shortcomings (e.g. Simons and Graham, 1999; Wang, 2001; Dori, 2001; Halpern, 2001; Hitz and Kappel, 1998; Glinz, 2000). However, there is still very little empirical evidence available describing the actual usage patterns or performance impacts of UML. There is a critical need for such empirical research to determine how UML is currently being used, how it is perceived by those individuals using it, and what individual, task and organizational factors are impacting its use.

A number of research models have emerged that attempt to explain the acceptance and utilization of technology. One such framework is provided by task-technology fit (TTF) theory (Goodhue and Thompson, 1995; Goodhue, 1998), which links performance with the fit between the task being performed and the particular type of technology being utilized. Researchers have used the TTF framework to investigate a wide assortment of information technologies, such as software maintenance tools (Dishaw and Strong, 1998), knowledge management systems (McCarthy, Aronson, and Mazouz, 2001), data warehousing (McCarthy, Aronson and Claffey, 2002), simulation modeling (Dishaw, Strong and Brandy, 2001), the World Wide Web (D’Ambra, 2001; Srivihok. and Burstein, 2000), e-commerce (Benslimane, Plaisent, and Bernard, 2003; Wells, Sarker, Urbaczewskiand Sarker, 2003), manufacturing task support systems (Tjahjono, Fakun, Greenough and Kay 2001), group support systems (Zigurs, Buckland, Connolly and Wilson, 1999; Murthy and Kerr, 2000; Shirani, Tafti, and Affisco, 1999), and enterprise resource planning (Smyth, 2001).

Assuming that we can learn to select technologies that are a better fit within the context of the organization, the research in this area has several important implications for managers planning enterprise wide strategy. The present study was conducted to explore how the adoption and usage of UML can be explained using the TTF framework.

METHODOLOGY

A review of the literature surrounding UML usage (Agarwal and Sinh, 2003; Cox, 2000; Dori, 2001; Erikson and Siau, 2003; Glinz, 2000; Halpern, 2001; Hitz and Kappel, 1998; Siau and Cao, 2001; Siau and Tian, 2001; Simons and Graham, 1999; Wang, 2001) led to the following questions:

1. Do individuals who use UML perceive it to be beneficial?
2. Does UML provide a task-technology fit to individuals who utilize it?
3. What are the characteristics that affect UML use?

The survey research instrument was developed utilizing constructs that were originally developed by Goodhue and Thompson (1995) and subsequently expanded by a number of other researchers (e.g. Goodhue, 1998; Dishaw and Strong, 1998; McCarthy, Aronson and Mazouz, 2001). The sample population for this survey consists of information technology professionals with experience utilizing UML for systems development.

The variables to be tested in this study are adapted from Goodhue’s (1998) task-technology fit instrument. They include:

1. Right data
2. Accuracy
3. Compatibility
4. Flexibility
5. Understandability
6. Level of detail
7. Training
8. Ambiguity

The research model used in this study is shown in Figure 1.
The survey questions were modified to reflect that the technology in question is UML and not information systems in general. The first part of the survey consists of UML usage questions mapped directly to the task-technology fit constructs as described by Goodhue (1998) with some modifications to make it UML specific. These questions were presented in a random order to avoid clustering by variable. Respondents were asked to indicate the answers to these questions using a Likert scale providing five possible levels of agreement (strongly disagree to strongly agree).

The second part of the survey contained questions which asked for additional information, divided into five subsections. The first subsection related to individual characteristics such as gender, educational background, and experience level. The second subsection dealt with project characteristics such as the type and complexity of application being developed. The third subsection focused on organizational characteristics, such as corporate culture and industry sector. Subsection 4 contained questions specifically relating to UML and how it is being utilized in the specific environment of the respondent.

The survey was administered as a pre-test pilot to a group of information technology students and faculty at a small, private university in the southeastern U.S. The purpose of the pilot was to obtain feedback on the overall readability of the instrument. The actual survey results from this group were not used as part of the data analysis.
The sample population used in this study was derived by accessing various online newsgroups with threads relating to UML, object-oriented analysis and design tools and software development methodologies (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 e-mail addresses of UML users were culled from the archives of these discussion groups as well as from other sources (e.g. UML related user groups, Web sites, conferences, and articles) and entered into a database of survey subjects. Targeting participants in this manner increased the chances that the population consisted only of those information technology professionals who have actually used UML on software development projects. Only those individuals believed to be serious users of UML, based on the context of the environment in which their name was encountered, were selected for the survey. Direct e-mails were sent to all of individuals in the database, asking them if they would agree to participate in a survey on UML usage and providing a link to the actual survey page.

Web-based surveys have become popular in recent years and have been validated in studies described by Schonlau, Fricker and Elliot (2002) who indicate that in many cases they are less time consuming, cheaper to conduct, and easier to execute. 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 represents a response rate of approximately 11%. Considering that no monetary incentives were offered to entice respondents to complete this survey, a practice typically used by commercial Internet researchers (Downes-Le Guin, Janowitz, Stone and Khorram, 2002), this response rate was considered to be a reasonably good outcome. The response rate in this study can be attributed in part to the interest level of the targeted group of UML users solicited. Based on the percentage of subjects who provided additional comments on the survey (28.2%) and who indicated a willingness to participate in future studies (58.8%), one may conclude that the individuals who did respond to this survey were relatively opinionated regarding UML usage and eager to express their views. Nineteen surveys were eliminated from the sample due to incomplete responses, leaving the final sample size used in this study at 131.

The present study consisted of 32 survey questions, mapped to eight variables. Variables were derived from the original task-technology fit study of Goodhue and Thompson (1995), with some minor modifications to reflect UML usage. Cronbach’s Alphas were computed for each of the eight variables. Reliability coefficients for each of the variables are shown in Table 1.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Number of Items</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right Data</td>
<td>4</td>
<td>.7732</td>
</tr>
<tr>
<td>Accuracy</td>
<td>4</td>
<td>.4446</td>
</tr>
<tr>
<td>Compatibility</td>
<td>4</td>
<td>.6003</td>
</tr>
<tr>
<td>Flexibility</td>
<td>4</td>
<td>.8175</td>
</tr>
<tr>
<td>Understandability</td>
<td>4</td>
<td>.7427</td>
</tr>
<tr>
<td>Level of Detail</td>
<td>4</td>
<td>.6280</td>
</tr>
<tr>
<td>Training</td>
<td>4</td>
<td>.7517</td>
</tr>
<tr>
<td>Ambiguity</td>
<td>4</td>
<td>.7584</td>
</tr>
</tbody>
</table>

Table 1 - Cronbach’s Alpha Reliability Analysis

Three of the constructs (accuracy, compatibility and level of detail) exhibited Cronbach’s Alphas below the accepted value of .70 and were therefore eliminated from the study. To determine the validity of the remaining constructs, a Pearson product-moment correlation was performed. Strong positive correlations at the 0.01 level of significance were exhibited for flexibility, right data, understandability and training. The ambiguity variable however, exhibited a negative correlation at the .05 level.
Prior to performing statistical analysis, all responses were normalized to account for the wording of the survey questions. For example, some questions were posed positively (e.g. “UML provides sufficiently detailed diagrammatic constructs”), while others were cast pejoratively (e.g. “I find UML to be too complex and difficult to understand”). After normalizing the data all numerical rankings were consistent, with 5 indicating the highest level of approval of UML and 1 indicating the lowest. Consistent with other TTF researchers (Goodhue and Thompson, 1995; Dishaw and Strong, 1998; McCarthy et al., 2002 and Tjahano et al., 2001), we were able to derive TTF indices for each dimension by averaging the scaled data. A cursory inspection of the obtained TTF indices (Table 3) indicates a response pattern that is slightly above neutral, which represents a generally positive perception of UML. Indices fell between 2.1 to 4.35, with an overall TTF index of 3.35.

### Table 3 - Distribution of TTF Indices

<table>
<thead>
<tr>
<th>N</th>
<th>Min. TTF</th>
<th>Max. TTF</th>
<th>Mean</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>131</td>
<td>2.10</td>
<td>4.35</td>
<td>3.3481</td>
<td>.4668</td>
</tr>
</tbody>
</table>

Breaking down the computed indices by TTF dimension (Table 4), reveals that perception of UML is by no means uniform across the variables used in this study. The ‘understandability’ dimension, for example, was found to have the highest index (3.89) while ‘ambiguity’ revealed a totally neutral index (3.0). As seen in Table 5, over 77% of the respondents in this survey had 4 or more years of experience in object-oriented technology. That, in conjunction with the high degree of college education of the respondents (87% of the population had at least a Bachelors degree - Table 6), may partially explain the relatively high index exhibited along this dimension. A major part of understanding UML is predicated on having a solid comprehension of the object-oriented model. Since most of our respondents were already highly experienced with this technology, it stands to reason that there would be little problem understanding the fundamental concepts underlying UML, which is based on that paradigm. Although UML has a reputation for being overly complex (Simons and Graham, 1999; Wang, 2001) we should also consider that most practitioners may only be using a subset of the language (Erikson and Siau, 2003), exhibiting the phenomenon known in software engineering as the ‘80/20 rule’ (i.e. 80% of the systems are developed using 20% of the language constructs). The relatively high ‘understandability’ index may actually reflect usage of a smaller subset of UML, which is less complex and easier to work with.

Several authors (Simons and Graham, 1999; Siau and Cao, 2001; Akehurst and Waters, 1999) have pointed to ambiguity as one of the more problematic aspects of UML. Ambiguity of UML constructs may indeed be related to the very way in which the standard was created, i.e. as a synthesis of already existing OOAD techniques. As if attempting to satisfy the widest range of participants involved in the standardization process, UML has incorporated a number of overlapping constructs, which potentially introduce ambiguity in their usage. The relatively low index calculated for the ‘ambiguity’ index in this study, may reflect this somewhat disjointed aspect of UML. Averages of scaled data for each of the TTF dimensions are displayed in Table 4.
It is interesting to inspect the company location of respondents, broken down according to TTF indices. Table 7 reveals that U.S. developers have a slightly lower overall TTF index than those in most of the other regions represented in the survey. Particularly noteworthy are the differences between the U.S./Canada, Europe and Asia/Middle East regions, each with sizable portions of the sample. Ironically, the level of satisfaction of U.S. developers seems to be lower than those in Europe and Asia. Perhaps this trend reflects the fact that non-U.S. developers have been using OOAD methods longer than those in the U.S. (Johnson and Hardgrave, 1999). Further investigation of the intercultural aspects of UML usage and adoption is warranted (Grossman, McCarthy and Aronson, 2003).
Table 7 - TTF Indices by Geographic Location of Company

<table>
<thead>
<tr>
<th>Geographic region</th>
<th>Frequency</th>
<th>Percent</th>
<th>Cumulative Percent</th>
<th>Aggregated TTF index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Africa</td>
<td>3</td>
<td>2.3</td>
<td>2.3</td>
<td>3.77</td>
</tr>
<tr>
<td>Asia / Middle East</td>
<td>17</td>
<td>13.0</td>
<td>15.3</td>
<td>3.52</td>
</tr>
<tr>
<td>Australia / New Zealand</td>
<td>5</td>
<td>3.8</td>
<td>19.1</td>
<td>3.15</td>
</tr>
<tr>
<td>Europe</td>
<td>52</td>
<td>39.7</td>
<td>58.8</td>
<td>3.39</td>
</tr>
<tr>
<td>South America</td>
<td>2</td>
<td>1.5</td>
<td>60.3</td>
<td>3.4</td>
</tr>
<tr>
<td>USA and Canada</td>
<td>52</td>
<td>39.7</td>
<td>100.0</td>
<td>3.24</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>131</strong></td>
<td><strong>100.0</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Implications**

There are a number of managerial implications that arise as we study UML in the context of task-technology fit. As UML’s popularity has increased, an entire industry providing related products and tools has also begun to emerge. Companies are starting to invest in UML, hoping that it will facilitate more productive software development. But UML is not a trivial technology. Simply throwing it at developers does not mean that it will result in the performance gains desired. We need to first ask a number of important questions to determine how well the usage of UML will fit within the organizational context. How will UML be used, i.e. casually or at a more rigorous level of detail? Do the users have adequate object-oriented experience to understand the complexities of UML and to be able to use it successfully? Does management support the use of UML and/or is there a champion in the company who can guide the effort? Is there adequate training available? These are just a few questions that need to be considered. In short, we need to take into account the multitude of complex factors that make up the organizational environment, striving to achieve alignment between them. The implication for management is clear. The greater the fit between the individual, task and technology, the better the chances that UML will be perceived positively thus leading to greater performance impacts. The model utilized in this study shows modest support for what is so intuitively obvious regarding task-technology fit. However, as can be seen by the somewhat lackluster perception of UML, there is still much about the technology that is open-ended and poorly defined to allow its users to perceive its benefit. As UML matures, it is tempting to conjecture that this situation will change and that a task-technology fit will be more definitively evident.

**Conclusion**

The characteristics identified in this study were right data, accuracy, compatibility, flexibility, understandability, level of detail, training, and ambiguity. Three variables were dropped from the model as a result of low reliability. These were ‘accuracy,’ ‘compatibility’ and ‘level of detail’.

One of Goodhue’s (1998) original constructs, ‘accuracy’ was meant to measure the correctness of the data. In the present study, ‘accuracy’ was altered slightly to reflect the ‘correctness’ of UML’s constructs. To some extent, it attempts to determine whether UML diagrams result in accurate depictions of the system, or whether they actually mislead the developer, as Simons and Graham’s (1999) ‘cognitive misdirection’ concept suggests. The questions in this category, although intuitively consistent, fail to hold together as evidenced by the low Chronbach’s alpha. One reason could be the experience level of the sample population, which was very high (approximately 77% of the respondents had 4 or more years experience with object-oriented technology). The concept of accuracy may not be relevant to this group of users, who already has a firm enough grasp of the object-oriented paradigm.

The ‘level of detail’ construct was also included in Goodhue’s (1998) questionnaire, and originally measured whether data in general was maintained at the right level of detail. In this survey it was meant to convey whether UML is easy to figure out or whether it is too detailed. Lack of cohesion within this variable might be related to the variety of ways in which UML is...
being used among those sampled. UML is used by some simply as a communications tool and by others as a formal mechanism for requirements definition and design (Fowler, 2004). To further complicate the matter, UML has fragmented into various dialects. Erikson and Siau (2003) make the point that there are separate UML domains for enterprise systems, Web-based systems, and real-time systems, which are evolving differently and which involve different levels of detail. We may also need to consider the process used along with UML (e.g. Unified Process, agile methodologies) as well, since some of these methodologies are inherently more detailed than others.

Like previous studies of other information technologies, this study suggests that task-technology fit theory has some explanatory power, but due to the elusive nature of UML, there are still many questions that need to be answered. While the aggregated index of 3.35 indicates a slightly positive perception of UML, it by no means represents an overwhelming endorsement. At this early stage in its evolution, it may be that the people using UML still do not have enough of a feeling for how this technology fits with the tasks they are trying to perform. The fact that respondents to this survey failed to express strong opinions may reflect the fact that UML is an immature standard that is still undergoing codification. The UML phenomenon presents an enigma. While it is increasingly being adopted throughout the world, there is really no consensus on how it should be used or on whether it is providing beneficial effects. Much of the literature points to a technology that is complex, poorly understood, and used inconsistently. Perhaps the results of this survey reflect a certain degree of confusion among the user community surrounding UML. Developers clearly seem eager to use this highly hyped technology which is making any real difference in the way they are performing their development tasks.

REFERENCES