ICIS 2010 Proceedings

International Conference on Information Systems (ICIS)

1-1-2010

UNDERSTANDING THE ADOPTION OF USE CASE NARRATIVES IN THE UNIFIED MODELING LANGUAGE

Brian Dobing
University of Lethbridge, brian.dobing@uleth.ca

Joerg Evermann
Memorial University of Newfoundland, jevermann@mun.ca

Jeffrey Parsons
Memorial University of Newfoundland, jeffreyp@mun.ca

UNDERSTANDING THE ADOPTION OF USE CASE NARRATIVES IN THE UNIFIED MODELING LANGUAGE

Completed Research Paper

Brian Dobing
University of Lethbridge
Lethbridge, AB, Canada
brian.dobing@uleth.ca

Joerg Evermann
Memorial University of Newfoundland
St. John's, NL, Canada
jevermann@mun.ca

Jeffrey Parsons
Memorial University of Newfoundland
St. John's, NL, Canada
jeffreyp@mun.ca

Abstract

This research examines the adoption of Use Case Narratives within the Unified Modeling Language (UML).Using the Technology Acceptance Model (TAM) as a framework, practitioners with UML experience were asked questions to measure their Perceived Ease of Use and Perceived Usefulness of Use Case Narratives and their Intentions to Adopt them. We extend Perceived Usefulness in the context of UML adoption to address the question "usefulness for what purpose(s)?" Generally, we find that TAM explains Use Case Narrative acceptance. More importantly, we find that Perceived Usefulness is explained by usefulness for specific software development tasks. This research provides three main contributions, beginning with an improved understanding of the role of Use Case Narratives in UML projects. Second, the study extends TAM by explaining how a technology is used rather than simply whether it is used. Third, this study provides a framework for future studies into other UML diagrams.

Keywords: Unified Modeling Language, Use Case, Use Case-driven, Technology Acceptance Model.

Abbreviation	Full Name	Comment
BI	Behavioral Intention	A construct of TAM
CASE	Computer Assisted Software Engineering	
MPCU	Model of Personal Computer Utilization	
OMG	Object Management Group	Organization managing the UML standard
PCI	Perceived Characteristics of Innovating	
PEOU	Perceived Ease of Use	A construct of TAM
PU	Perceived Usefulness	A construct of TAM
SDM	Systems Development Methodology	
TAM	Technology Acceptance Model	
TPB	Theory of Planned Behavior	
UML	Unified Modeling Language	

Introduction

The Unified Modeling Language (UML) is widely viewed as the de facto standard for object-oriented systems analysis and design. A survey by Zeichick (2004) found that 20% of organizations were using it for all projects and a further 58% were using it for at least some. When asked about their future plans, 42% said they were planning to use the UML for modeling in all future projects and an additional 40% plan to use UML for modeling in some, but not all, projects.

Notwithstanding this interest in the UML, there has been little research into how the language is being adopted and applied in software development. Among UML adopters, there is evidence of selective adoption of parts of the language (Dobing and Parsons 2006; Grossman et al. 2005; Zeichick 2002). This may be a form of Agile Modeling or a coping mechanism for less experienced practitioners (Dobing and Parsons 2006), but little is known about how these selective adoption decisions are made.

In this paper, we extend prior research on technology acceptance to the UML context. As a modeling language for systems development, the UML is properly considered a technology. In view of the complexity of UML – with its many diagram types and the large number of modeling constructs within each diagram type (Kobryn 1999; Siau and Cao 2001) – and its selective adoption, the notion of acceptance of the UML as a whole might not be a useful construct. In this study, we focus on the acceptance of Use Case Narratives rather than of the UML itself, as proponents often promote a "Use Case-driven" approach to UML use. Thus, this study differs in important ways from most technologies previously studied. First, the study examines how a technology is used rather than simply whether it is used. Second, the usefulness of this technology is affected by team or organizational practices; system development is not an individual activity. Finally, the usefulness of the language must be evaluated as a complex phenomenon with multiple aspects, such as usefulness with general development tasks (e.g., communication with clients, systems design and testing) and in supporting the development of other diagrams (e.g., transition from Use Case Narratives to a Class Diagram).

In the following sections, we briefly review the literatures on UML and Use Case Narratives as well as on technology acceptance in general and specifically among information technology (IT) professionals, present a research model regarding the acceptance of Use Case Narratives and the Use Case-driven approach when using the UML, test this model and discuss its implications.

Use Case Narratives in the UML

Use Cases were introduced by Jacobson et al. (1992) to specify "the functionality the system has to offer from a user's perspective" (p.157). In other words, they specify the functional requirements but typically not the non-functional requirements such as performance (e.g., number of users that can be handled simultaneously, transaction response times, etc.) and security. Use cases are typically written in text format although they can be supplemented with flow charts, decision trees, etc. Cockburn (2001) remains one of the most popular guides on how Use Cases should be written. There is also an accompanying Use Case Diagram (Object Management Group 2009a, pp.585-603; Rumbaugh et al. 2005, pp.34,77-80,668-677) that provides an overview of Use Cases showing how "actors" (e.g., customers, employees, other systems) interact with each Use Case. Our focus is on the text Use Case (referred to here as the 'Use Case Narrative' to avoid confusion), rather than on Use Case Diagrams that provide an overview of Use Case Narratives and establish system boundaries and scope.

When UML Version 1.1 was adopted by the Object Management Group (OMG) in 1997 (Object Management Group 2009b, p.9), Use Cases were included "to capture the requirements of a system, that is, what a system is supposed to do" (Object Management Group 2009a, p.585). The UML itself "is a general purpose language, that is expected to be customized for a wide variety of domains, platforms and methods" (Object Management Group 2009b, p.207). Nevertheless, from the first books on the subject, the UML has frequently been associated with object-oriented system development and a Use Case-driven approach (e.g., Bahrami 1999; Booch et al. 1999; Harmon and Watson 1998; Larman 1998). For example, Rumbaugh et al. (2005, p.xiv) state that the UML is "designed to support an iterative, incremental, use-case-driven process" and this approach is reflected through most of the UML literature. Rosenberg and Scott (2001) even titled their book *Applying Use Case Driven Object Modeling with UML*. Jacobson et al. (1999, p.37) define this approach as one in which Use Cases "drive the whole development process since most activities such as analysis, design, and test are performed starting from the use cases. Design and test can also be planned and coordinated in terms of use cases." A Use-Case driven approach ensures that system development is "requirements driven" and some prefer that broader term because Use Cases are not intended to capture all requirements (e.g., Ambler 2002, p.304).

While Use Case Narratives and the Use Case-driven approach are supported by much of the UML literature, there are some exceptions. Anderson (1999a) states that, "You cannot have good User Interface Interaction Design and Usability together with Use Cases." Anderson (1999b) provides some further criticisms in a subsequent article, ominously entitled "Use Cases still considered Dangerous," suggesting "they can lead to poor definition of classes" and problems with exception handling. Glinz (2000) identifies some specific requirements that Use Case modeling doesn't seem able to capture. Furthermore, critics note that the format of Use Case Narratives is not part of the UML specifications. The only direction provided is that "use cases are typically specified in various idiosyncratic formats such as natural language, tables, trees, etc." (Object Management Group 2009a, p.590).

Use Cases can also be misused; determining how many Use Case Narratives to write and in how much detail is a problem (Dobing and Parsons 2000). Fowler (1998) suggests than improper decomposition, excessive abstraction and reliance on prototype interfaces (which hide the processing that must go on to make them work) can also cause problems when working with Use Cases.

To our knowledge, only a few researchers have examined acceptance of the UML and none has addressed acceptance of Use Case Narratives and the Use Case-driven approach. Grossman et al. (2005) applied the Task-Technology Fit model in their study of the UML and measured perceptions of "those directly involved with UML" (with their mean values on a five-point scale, from Strongly Disagree to Strongly Agree, shown in parentheses): Right Data (3.17), Flexibility (3.53), Understandability (3.89), Training (3.15), and Ambiguity (3.00). These results do not show strong support for the UML.

More research on acceptance of the UML is clearly needed and some must be done at the diagram and construct levels. (There is also a larger and complementary literature that works in the opposite direction, identifying modeling needs and then suggesting ways to improve the UML to meet them.) Three UML surveys (Dobing and Parsons 2006; Grossman et al. 2005; Zeichick 2002) have all found that most respondents were not using all the diagrams provided in UML 1.5. Of particular interest are Use Cases and the Use Case-driven approach. Although both have been widely endorsed in the literature since the inception of the UML, Dobing and Parsons (2006) found that only 44% of respondents reported that Use Case Narratives were written for at least two-thirds of their projects. The current research was undertaken to help further understand the acceptance of the Use Case-driven approach and the reasons for its use.

Technology Acceptance Model in IS Research

Technology Acceptance in General

The most common model of technology adoption behavior is the Technology Acceptance Model (TAM), introduced by Davis (1986; 1989). In its initial form, TAM proposed that acceptance of a technology should be positively related to its perceived benefits, operationalized as Perceived Usefulness (PU), and inversely related to its perceived adoption cost. Because acquisition costs are typically borne by the organization, the primary cost to individuals within the organization is the effort required to learn and use the technology. This is operationalized through Perceived Ease of Use (PEOU) which is inversely related to cost, so PEOU was hypothesized to be directly related to Acceptance. Acceptance is usually operationalized as one or more of Attitude towards the technology, Behavioral Intention (BI) to adopt it, or Use.

There is an extensive body of empirical research supporting TAM, much of it based on acceptance of PC-based personal productivity technologies, such as email (e.g., Adams et al. 1992; Burton-Jones and Hubona 2005), word processors (e.g., Adams et al. 1992; Chau 1996a; Davis et al. 1989), databases (e.g., Mathieson et al. 2001; Strong et al. 2006; Thompson et al. 2006), the Internet (e.g., Agarwal and Prasad 1997) and graphics packages (e.g., Adams et al. 1992; Davis 1989; Venkatesh and Davis 1996). Often, the subjects had previously used little or no computer-based technology.

In their meta-analysis of the TAM literature, King and He (2006) found that PU explains BI better than PEOU, with the major effect of PEOU being on PU rather than directly on BI. The direct path from PEOU to BI is even lower for "professional" than "general" users. They also examined the many attempts to extend TAM such as TAM2

(Venkatesh and Davis 2000), which found that Subjective Norm, Image, Job Relevance, Output Quality and Result Demonstrability were all significant prior factors while Experience and Voluntariness significantly mediated the Subjective Norm-Behavioral Intention relationship. Other variables that have been proposed to extend TAM include those related to implementation approaches (training and available resources), subject characteristics (type, organization level or seniority, gender, age, education, and self-efficacy) and compatibility with prior technologies.

Other studies have focused on specialized user groups, such as physicians (Chau and Hu 2001; 2002), senior executives (Pijpers and van Montfort 2006), and knowledge workers (Venkatesh et al. 2002). Some differences were found. For example, while acceptance of telemedicine by physicians was found to be largely determined by Usefulness rather than Ease of Use, use of Executive Information Systems by senior executives depended almost equally on both factors.

Technology Acceptance by IT Professionals

While IT professionals might be expected to adopt new technology faster than other groups, this is not always true. For example, Hardgrave (1995) found that the percentage of respondents using prototyping increased from 33% (citing Langle et al. 1984) to only 71% a decade later. Iivari (1996) provides a good summary on Computer Assisted Software Engineering (CASE) tool usage, finding not only relatively low use by organizations but high rates of non-use even after organizations have purchased them. Fitzgerald (1998) notes that "practitioners have been somewhat slow in adopting SDMs" (System Development Methodologies); his survey found 60% of organizations were not using a formal SDM (either internal or commercial). Findings such as these have led to studies on a number of technologies used in systems development, including object-oriented programming languages (e.g., Agarwal and Prasad 2000; Fichman and Kemerer 1997), CASE tools (Chau 1996b; Dishaw and Strong 1999), software maintenance tools (Dishaw and Strong 1998; 2003), system development methodologies (Hardgrave et al. 2003; Roberts et al. 1999; Roberts et al. 1998), and object oriented methodologies in particular (Hardgrave and Johnson 2003; Lee et al. 2006; Riemenschneider and Hardgrave 2001; Riemenschneider et al. 2002). These studies have used a variety of theoretical foundations, including TAM, TAM2, Task-Technology Fit (Goodhue 1998; Goodhue and Thompson 1995), Theory of Planned Behavior (TPB) (Ajzen 1988), Perceived Characteristics of Innovating (PCI) (Moore and Benbasat 1991), and the Model of Personal Computer Utilization (MPCU) (Thompson et al. 1991).

There are several possible explanations for resistance to new technologies among IT professionals. Often early versions of new technologies are not nearly as useful or easy to use as those that come later. Because of the adoption cost, it is tempting for organizations to wait until a technology has been proven. Another possible explanation is that, relative to general office software, adopting new system building technologies also requires a greater organizational commitment. This is partly due to the generally high cost of acquiring such technologies and the often higher cost of training people to use them. Changing methodology is also very likely to be accompanied by a change in CASE tools (or other supporting software) and possibly a change in programming language. So the impact of such a change can be considerable.

Another difficulty facing IT professionals is that they cannot easily experiment with new development methodologies on an individual basis, which Rogers (1983) refers to as low "trialability." The UML and Use Case Narratives are both examples of this. To properly determine their value in a Use Case-driven context, a developer must not only write Use Case Narratives but discuss them with clients and team members, use them to develop Class and other Diagrams, and build testing programs around them. Thus, adoption of the UML and Use Cases needs be at least a team level decision and more likely one to be made by senior management in the IT Department. In sum, a compelling case can be made that the adoption of systems development technologies by IT professionals is a more complex phenomenon than adoption of personal productivity software by individuals.

Research Model

Figure 1 shows the general research model, which is based on the original TAM (Davis 1989) and is generally consistent with both the King and He (2006) meta-analysis and the literature on acceptance of technology by IT professionals. This section reviews the relationships in this model. Perceived Usefulness (and similar constructs such as Relative Advantage) have been shown to be a significant and important determinant of Acceptance of COBOL/CICS maintenance support tools (Dishaw and Strong 1998; 2003), software development methodologies (Hardgrave et al. 2003; Hardgrave and Johnson 2003; Riemenschneider and Hardgrave 2001; Riemenschneider et al.

2002), and object oriented technology (Lee et al. 2006). Acceptance can be measured as Attitudes, Intentions, or Use; which is more appropriate depends on the subjects, technology and research design. We are focusing on Attitudes and Intentions (both for self and others) as Use is typically not an individual decision for systems developers. Our research instrument appears as an Appendix.

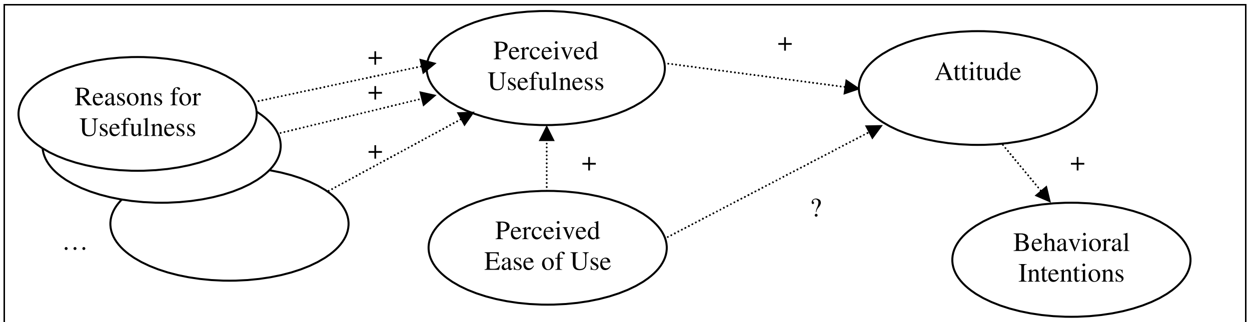

Figure 1. Research Model (based on the Technology Acceptance Model)

As noted earlier (King and He 2006), the measured relationship between PEOU and Acceptance is usually weak and tends to be weaker when the subjects have been professionals (generally not from the IT area). Among studies involving IT professionals, some support a positive relationship from PEOU to Acceptance (e.g., Chau (1996b) studying CASE tools and Riemenschneider & Hardgrave (2001) studying System Development Methodologies), but most others do not. We have therefore marked this relationship with a "?" to indicate these different views and have not hypothesized whether the PEOU to Acceptance relationship will be part of the final model or not.

Chau (1996b) also found that Ease of Use had a greater direct effect on Acceptance of CASE tools than Usefulness, but this was not supported in other studies. However, Ease of Use is positively related to Usefulness in the King and He (2006) meta-analysis and in several studies on IT professionals (e.g., Chau 1996b; Dishaw and Strong 1999; Lee et al. 2006; Riemenschneider and Hardgrave 2001). Therefore, we expect the PEOU to PU relationship to be supported in this study.

More recently, there have been suggestions that TAM research needs to move in some new directions to remain relevant and interesting. Benbasat and Barki (2007) state that "TAM-based research has paid scant attention to the antecedents of its belief constructs: most importantly, IT artifact design and evaluation." This comment is echoed by Bagozzi (2007); "almost no research has deepened TAM in the sense of explaining PU and PEOU." To address this concern, the basic TAM model has been extended here by proposing a set of Reasons for Usefulness. Specifically, Use Case Narratives are intended for use for several tasks in systems development, and we expect perceived usefulness for these tasks to be positively related to overall PU. We believe this approach is consistent with the conceptualization of usage proposed by Burton-Jones and Straub (2006). This is explained in more detail next.

Research Design

We used a Web-based survey to conduct this research. This approach is economical, allowing us to reach a large number of potential respondents, and seems to achieve a reasonable response rate from IT professionals (Dobing and Parsons 2006; Grossman et al. 2005; Zeichick 2002). We contacted the OMG, the organization responsible for the UML standard, who sent emails to its members. We also emailed respondents from an earlier survey, invited members of related online groups to participate, and asked related web sites (including the OMG) to post links on their pages. No paid advertising was used. While this approach does not provide the same level of control as a targeted survey, the responses are expected to be much more diverse than could be obtained by approaching a few organizations or local IT associations. As Huisman and Iivari (2006) noted, different stakeholder groups "perceive the benefits and problems of systems development technologies differently." Thus, diversity of roles is important in addition to obtaining responses from many organizations and countries.

In addition to the research model constructs, the survey asked respondents about their organization, experience with system development in general and their use of the UML and Use Case Narratives. Respondents with no experience in system development or with the UML were not asked to complete the remainder of the survey. Respondents without Use Case Narrative experience were able to complete the survey. This is consistent with prior TAM studies in which respondents had limited, if any, hands-on experience with the technology in question.

Acceptance

Consistent with many prior studies, we measured acceptance in terms of Attitude and Behavioral Intention. However, the latter can be problematic because adoption decisions may be made at the organizational level. Respondents may have no intention of adopting Use Case Narratives simply because their organizations have no such intention. So Intention items were qualified with phrases such as "Given the opportunity." In addition, we distinguished between intention to adopt Use Case Narratives oneself (which might be dictated by organizational policies) and intentions to recommend the adoption of Use Case Narratives to others. The Attitude items are written to ask about respondent beliefs about writing Use Case Narratives, which can be different from those of their organization.

Perceived Usefulness

Perceived Usefulness is a critical independent variable; past TAM studies have shown strong support for the link between Usefulness and Acceptance. Generally these studies have used measures of Usefulness that are largely domain independent. Very similar items have been used to measure the Perceived Usefulness of spreadsheets, email, web searches, etc. But such items provide little insight into why people view these technologies as useful.

Reasons for Usefulness

This study focuses on understanding why Use Case Narratives are perceived as Useful (or not). Ivar Jacobson, the originator of Use Cases, points to many of their advantages in his Forward to Armour and Miller (2001, xiii-xiv). Table 1 lists the key advantages he identifies along with a few others.

Not everyone has accepted all of Jacobson's points as valid. While there is general support for the view that Use Cases help identify classes, Meyer (1997, p738) states that "Use cases are not a good tool for finding classes." Ambler (2002, p.304) states that "[Use Cases] aren't very good at documenting … user interface requirements." In a survey of UML practitioners (Dobing and Parsons 2008), respondents agreed. To some extent, this disagreement reflects a difference in interpretation. Use Cases can identify the interfaces that are needed (e.g., a way to add or modify customer data) but do not normally specify all the data fields required and certainly don't specify entry controls, layouts, colors, etc. User Interface Design has been included but is not expected to make a major contribution to the Perceived Usefulness of Use Cases.

Arlow and Neustadt (2004, p.92) argue that programmers find Use Cases less useful because they don't understand the user domain and the terminology. But some programmers may acquire the necessary background. So Programming has also been included, which is consistent with the Use-Case driven approach, but with expectation that the relationship with Perceived Usefulness could be low or even negative.

Although not mentioned by Jacobson, maintenance can now be up to 90% of the total cost over the life of a system (Seacord et al., 2003, p.5). Moreover, much of that maintenance is likely to be done by people not involved in the original development and thus needing to understand what the system is supposed to do. Using contract professional developers rather than students, Dzidek et al. (2008) found a "54 percent increase in the functional correctness of changes" by those with access to UML documentation. While only Use Case Diagrams (not Narratives) were used in their experiment, this aspect of software development merits more attention.

Finally, the reason for using any artifact in system development is to improve effectiveness (software quality) and/or efficiency. These are also included in the model.

Table 1: Potential Reasons for the Usefulness of Use Case Narratives

Construct	Rationale[1]	Additional Literature
Communication	Use cases are the means to communicate with end users and customers	Ashrafi and Ashrafi (2009, p.183); Pender (2003, p.402)
Testing	Use cases are the base for integration testing	Armour and Miller (2001, p.233); Ashrafi and Ashrafi (2009, p.183); Jacobson et al. (1999, p.5)
	Use cases define test cases	
Programming		Arlow and Neustadt (2004, p.92); Jacobson et al. (1999, pp.5,34)
Project Management	Use cases help us with estimation of project size and required resources	Dennis et al. (2005, pp.186-190); Martin, Biddle, Noble (2004)
	Use cases are a tool for controlling a project	
	Use cases drive the development activities	
Efficiency		
Maintenance		
Improving Quality		
Developing Class Diagram[2]	Use cases are the base for object derivation; objects naturally fall out of use cases	Bahrami (1999, pp.46, 128); Harmon & Watson (1998, p.112;Meyer (1997, p.738)
Developing User Interfaces	Use cases are the base for user interface and experience design	Ambler (2002, p. 304); Dobing and Parsons (2006)

[1] All rationales are direct quotes from Armour and Miller (2001, xiii-xiv)
[2] To keep the survey to a reasonable length, this study did not inquire about the usefulness of Use Case Narratives in creating other UML diagram types.

Perceived Ease of Use

One reason why people may not use a technology they perceive as useful is that they believe the technology is difficult to learn and/or use. This seems a reasonable concern for many of the technologies which have been studied, but may not apply well to Use Case Narratives. Writing a single Use Case should be a relatively straightforward task. However, Use Case writing may be perceived as more difficult if there are very strict guidelines imposed on how they are to be written. Encouraging highly detailed Use Cases could also increase perceived difficulty as they would take longer to write, but in another sense this could simplify the task for beginners who would no longer need to make judgments on what to include and exclude. Creating a complete set, even with relatively agile approaches, requires some level of consistency in style and terms used.

Nevertheless, it is difficult to argue that Use Case writing is likely to be seen as difficult when compared to the technologies IS professionals have faced in past TAM studies (including CASE tools, new programming languages and new methodologies). One Ease of Use item refers to "creating and organizing Use Case Narratives" to capture the larger scale issues.

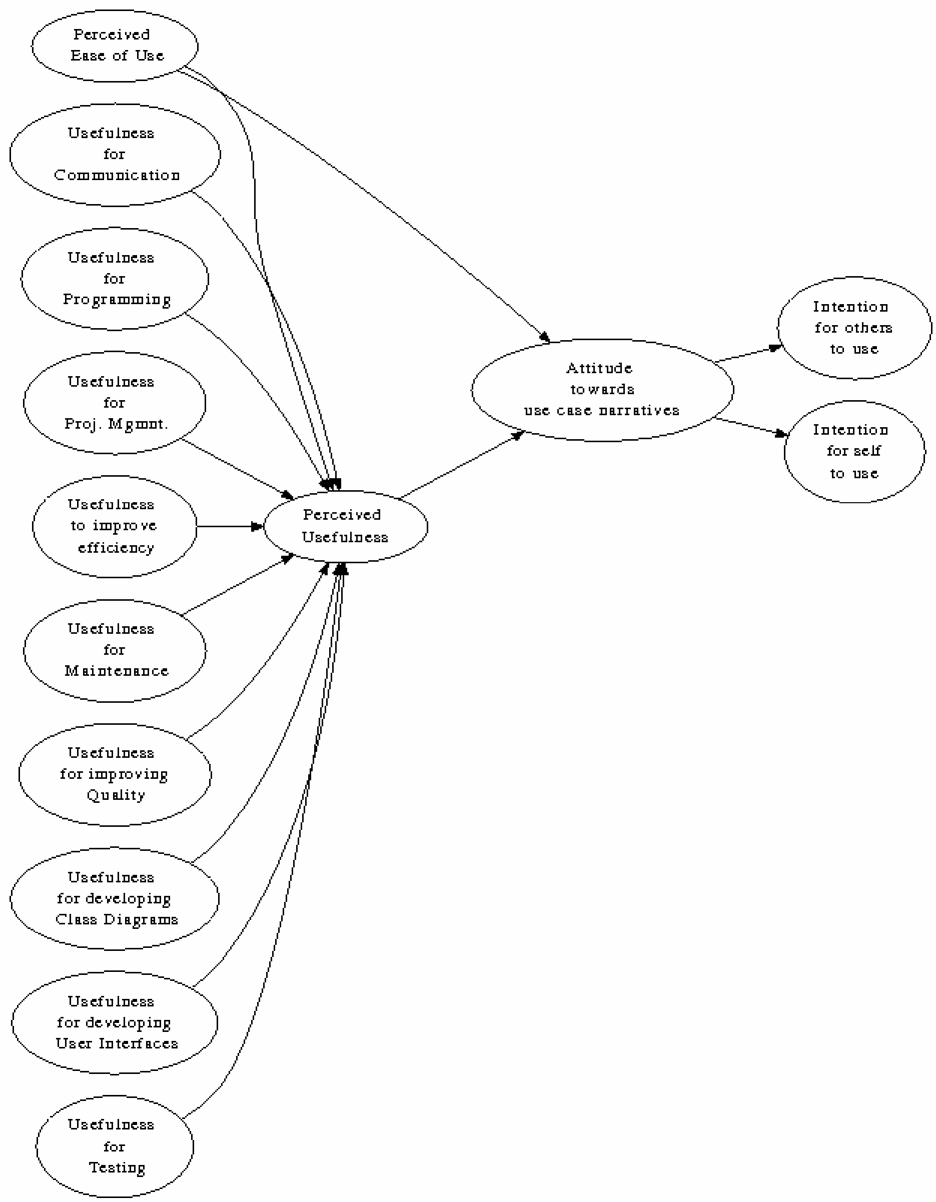

Figure 2. Extended Research Model (incorporating antecedents to Perceived Usefulness)

Results

The web survey was opened in June 2007. The set of items used for this study is shown in the Appendix. The results reported here are based on data collected until January 2010. There were 1891 hits on the site, with the majority (1380) providing no response at all. For the survey items analyzed here, 205 responses were found to be relatively complete. Eight of these responses were excluded due to low variance across the items (e.g. all 5's). Nine responses were excluded because of duplication (based on the provided e-mail address). We examined the extent of missing data and discarded responses that did not provide data on at least one item for each construct. We also discarded four responses for which the item score was more than three standard deviations beyond the average item score (univariate outliers). In total, we discarded 32 responses. The result was a data set with 173 responses. Only seven IP addresses were used twice, and none more often than twice.

Respondents reported playing a variety of roles, with most working as Business/Operation Analysts (44%), Architects/Designers (24%), Senior Managers/Project Leaders (13%) or Testers (8%). When asked about typical application areas, the responses were much more diverse with 24% choosing E-Commerce/Web, 19% Administrative and 8% Customer Relationship Management, while 36% picked “Other.” Both questions allowed only one response (radio buttons rather than check boxes) but they did provide an opportunity for comments. Responses came from IP addresses of at least 33 different countries, with 12 having at least three responses. (The European Union had nine responses that cannot be separated by country.) The United States was the source of 40%; Germany was second at 6%. Descriptive statistics of the sample are shown in Table 2 below.

Table 2: Sample Descriptive Statistics

	1st quartile	Median	3rd quartile
Completed projects	5	8	15
Completed UML projects	3	5	10
Completed Use case projects	3	5	10
Use case experience (1=low-5=high)	3	4	5
Direct use case experience (1=low-5=high)	3	4	5
Team UML experience	2	3	3
Project Size			
Budget (USD equivalent)	475,000	1,000,000	5,000,000
Person-years	3	10	20
Lines of code (irrespective of programming language)	2000	100,000	500,000
Number of use cases	20	40	100
Number of classes	40	100	266.2

We used partial least squares (PLS) for fitting our model to the data, using the plspm package in the R system (Sanchez and Trinchera 2009). Because list-wise deletion of missing values may lead to significant biases if the data is not MCAR (missing completely at random), we chose to use imputation, based on the Amelia package in R (Honaker et al. 2010). We used bootstrapping with 200 re-samples to determine significance of parameter estimates. The model that was fitted is shown in Figure 3. Table 3 below provides a summary of the estimation results.

While there is no overall test of model fit in PLS, there are a variety of heuristics intended to ensure adequate validity and reliability of the measurement model. Chin (1998) suggests that the r^2 values for endogenous latent variables are one criterion for assessing the structural model. Generally, the higher the r^2, the better the predictive ability of the model (Goetz et al. 2010). Given our model structure, 72.5% of the variance of overall perceived usefulness is explained, leading to 67.5% explained variance for users' attitude towards Use Case Narratives. This attitude in turn explains 60% and 70% of the variance in the intentions to adopt Use Case Narratives (for others, for oneself, respectively).

Table 3: PLS Estimation Results and Latent Variable Correlations

Latent Variable	Formative /Reflective	# of Items	r^2	Reliability (α)	Internal Consistency (ρ)	AVE	Latent Variable Correlations												
Ease of Use	R	4		.71	.82	.53													
Usefulness for Communication	R	3		.78	.87	.69	.53												
Usefulness for Programming	R	3		.80	.88	.72	.39	.46											
Usefulness for Project Management	F	4					.56	.58	.45										
Usefulness for Improving Efficiency	R	3		.73	.85	.64	.54	.66	.56	.64									
Usefulness for Maintenance	F	3					.57	.67	.59	.62	.66								
Usefulness for Improving Quality	R	3		.78	.87	.70	.56	.74	.55	.69	.70	.72							
Usefulness for developing Class Diagrams	F	4					.39	.47	.41	.46	.49	.56	.59						
Usefulness for developing User Interfaces	R	3		.87	.92	.79	.43	.41	.36	.39	.33	.49	.44	.48					
Usefulness for Testing	R	3		.90	.94	.84	.52	.48	.49	.58	.43	.59	.65	.48	.41				
Perceived Usefulness	R	3	0.73	.87	.92	.79	.65	.70	.54	.63	.58	.64	.79	.54	.42	.62			
Attitude towards Use Case Narratives	R	3	0.68	.82	.89	.74	.61	.58	.47	.58	.61	.62	.71	.55	.36	.56	.81		
Intentions for Self to Use	R	2	0.70	.84	.92	.86	.64	.69	.53	.64	.64	.65	.79	.54	.39	.58	.85	.84	
Intentions for Others to Use	R	2	0.60	.88	.94	.89	.59	.71	.53	.67	.62	.68	.76	.51	.37	.55	.80	.78	.85

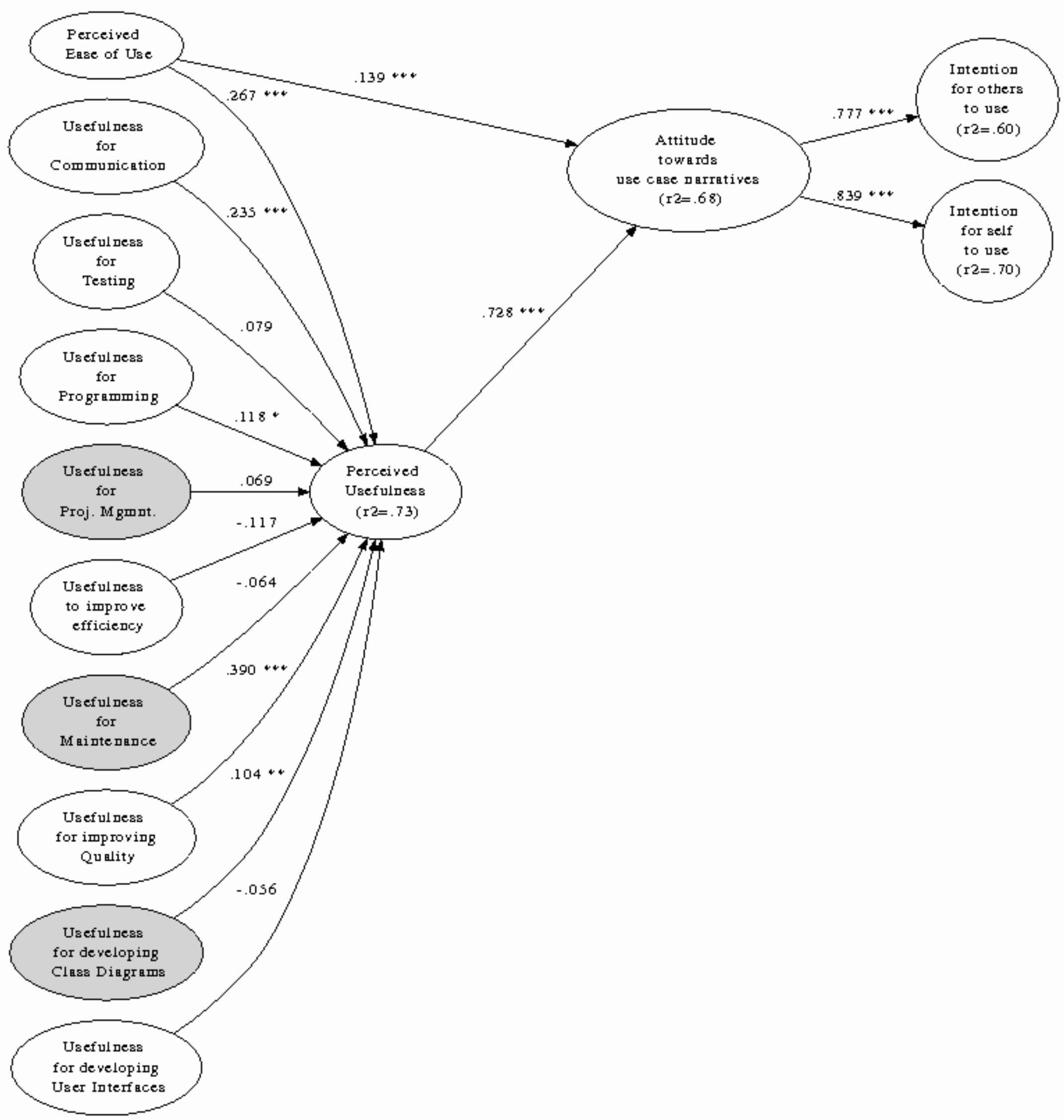

Figure 3: PLS Model Fitted to Data

Notes: Shaded concepts are formative. Paths show mean bootstrap values for coefficients.
Asterisks indicate significance levels (* - .10; ** - .05; *** - .01).

The average variance extracted (AVE) of a latent variable can also be used to assess the quality of models (Chin 1998). The AVE of each latent variable should be greater than 0.5 (Chin 2010; Chin 1998; Gefen and Straub 2005) and its square root should be greater than the correlations of that latent variable with other latent variables (Chin 1998; Chin 2010; Fornell and Larcker 1981; Gefen and Straub 2005; Hulland 1999). A more conservative heuristic is proposed by Gefen et al. (2000) who suggest that the AVE, rather than its root, should be greater than correlations with other latent variables. Finally, Fornell and Larcker (1981) argue that the AVE of a latent variable should be larger than its r^2. Our model satisfies most of these criteria. The root of the AVE for each variable is greater than the inter-variable correlations for all variables, and for Usefulness for Programming, Usefulness for Developing User Interfaces, Usefulness for Testing, Perceived Usefulness and the two intention variables, the AVE is greater than inter-variable correlations.

Another measure of model quality, based on factor-analytic considerations, is construct reliability. This can be assessed using either Cronbach's α (Goetz et al. 2010; Gefen et al. 2000; Straub et al. 2004) or using the composite reliability or internal consistency metric ρ (Goetz et al. 2010; Chin 1998; Chin and Gopal 1995; Fornell and Larcker 1981). As a guideline, construct reliability should exceed 0.7 (Hulland 1999; Gefen et al. 2000; Straub et al. 2004) or 0.6 (Goetz et al. 2010). For our fitted model, both α and ρ for all variables satisfy this criterion.

Other criteria based on factor-analytic considerations suggest that the loadings of items on their latent variable should exceed 0.707, so that the latent variable accounts for more than 50% of the item variance (Goetz et al. 2010; Hulland 1999; Gefen et al. 2000; Chin and Gopal 1995; Straub et al. 2004). On the other hand, the loadings of items on latent variables for which they are not indicators should be comparatively low (Chin 2010; Goetz et al. 2010; Hulland 1999; Gefen et al. 2000; Gefen and Straub 2005; Straub et al. 2004). Gefen and Straub (2005) suggest a relative criterion whereby item loadings should be higher than cross-loadings by at least 0.1. All items in our model have factor loadings > 0.7, many of them > 0.8, except for one item of Usefulness for Project Management with a loading of 0.561. Cross-loadings on all but four items satisfy Gefen and Straub's (2005) relative criterion. Finally, recognizing the need for an overall measure of model quality, goodness-of-fit (GoF) metrics (absolute, relative, measurement and structural GoF) have been proposed that take into account the PLS optimization objective and measure the "achievable fit" (Tenenhaus et al. 2004; Esposito Vinzi et al. 2010). While no guidelines as to acceptable values are offered, the authors seem to imply that their examples, which achieved a relative GoF of 0.78, are acceptable. Our fitted model had an absolute goodness-of-fit of 0.6991, and a relative goodness-of-fit of 0.8693. The GoF measures for the structural part (0.9851) and the measurement model (0.7672) both indicate acceptable goodness-of-fit.

Having established validity and reliability, we examined the hypothesized effects for significance. All items were measured on 5-point Likert scales, so that path coefficients are immediately comparable. The largest positive contribution to overall Perceived Usefulness comes from the usefulness of Use Case Narratives for improving Software Quality, followed by the Perceived Ease of Use. The high path coefficients between Perceived Usefulness, Perceived Ease of Use, Attitude and Behavioral Intentions, together with the comparatively high r^2 values, confirm the validity of the generic TAM model.

Discussion

The results of this study indicate that the adoption of Use Case Narratives can be explained using TAM. In particular, consistent with prior research we found that the general relationships found in the King and He (2006) meta-analysis also held here. Perceived Usefulness explained Attitude better than Perceived Ease of Use, with PEOU having a stronger effect on PU than on Attitude. The key contribution of this research is to provide a deeper understanding of the meaning of Perceived Usefulness of Use Case Narratives. Specifically, we find that Perceived Usefulness of Use Case Narratives can be understood in terms of usefulness for specific systems development tasks. Significant positive relationships were found for Communication, Programming, Improving Software Quality, and Developing Class Diagrams. The inclusion of Programming runs counter to the views of Arlow and Neustadt (2004) cited earlier that programmers have difficulty understanding Use Case Narratives. Perhaps programmers see them as useful despite that difficulty or that many of them have sufficient involvement in the requirements determination phase so that they do understand the Use Cases.

There are also positive but non-significant relationships between Usefulness for Testing and Usefulness for Project Management and Perceived Usefulness. Many books, including Jacobson et al. (1999, p.5), Armour and Miller (2001, p.233) and Ashrafi and Ashrafi (2009, p.183), suggest that Use Cases should be an important source of test cases or scenarios, which represent different paths through a Use Case. Systems should be tested against the requirements. But, as Mills (2002) notes, there can be problems doing this in practice, particularly if Use Cases are unclear, inconsistent, ambiguous or incomplete. Similar concerns may affect how well Use Cases can support Project Management, although some studies (e.g., Martin et al. 2004; Ludlow 2007; Periyasamy and Ghode 2009) suggest they are useful for this purpose.

In contrast, we find negative, but non-significant, relationships between Usefulness for Efficiency, Maintenance, and Developing User Interfaces and Perceived Usefulness. These results are consistent with a view in which the effort required to create (Efficiency) and manage (Maintenance) Use Case Narratives detracts from their overall usefulness. Also, those responsible for maintaining a system are often different from those who developed it and thus they may not have the domain knowledge to understand the Use Cases. While Ratanotayanon et al. (2009)

found that "user stories" (which have similarities to Use Case Narratives) can be useful in software maintenance, their study was based on a tool that "capture[d] links between user stories, test cases, and source code." Finally, the lack of a positive relationship between Usefulness for Developing User Interfaces and Perceived Usefulness is consistent with the role of Use Case Narratives, which only identifies where interfaces might be needed rather than providing guidance on their design, and with prior research indicating that support for developing user interfaces is a key UML deficiency (Dobing and Parsons 2006).

An additional contribution is the decomposition of usage intentions into two components – intentions to adopt for oneself and intentions to encourage others to do so. Intentions to adopt modeling techniques in information systems development can be conceptualized differently than intentions to adopt technologies for individual use, since decisions to use such technologies are often made at organizational or team levels. Our results support a conceptualization that distinguishes these types of intentions.

Collectively, these results suggest that the acceptance of Use Case Narratives can be explained by traditional TAM variables such as Perceived Ease of Use and Perceived Usefulness. More importantly, we argue and provide evidence that Perceived Usefulness needs to be understood in terms of usefulness for specific information systems development tasks, consistent with the conceptualization of usage proposed by Burton-Jones and Straub (2006). Moreover, this research points to the need to understand acceptance of a complex technology such as UML, not as a single technology, but in terms of its constituent modeling techniques (including Use Case Narratives) and the tasks for which they are used.

This study can serve as a starting point for future research to examine the acceptance of UML and its component modeling techniques (diagram types and constructs within those diagram types). Prior research has shown varying levels of use of the different UML diagram types (Dobing and Parsons 2006). For example, as Class Diagrams are the most frequently used component of the UML, it would be useful to examine factors that influence intentions to use these diagrams. By examining diagram adoption intentions, it will be possible to develop a better understanding of how the usefulness of each diagram type can be explained in terms of usefulness *for specific systems analysis and design tasks*. One interesting avenue for further research is to explore whether these tasks are inherently domain-specific, or whether a general conceptualization can be developed that can span technology adoption contexts beyond software development.

Summary

This paper examines how system developers are using Use Case Narratives, an important part of the UML. The study expands on previous research into adoption of IS development technologies among IS professionals by considering issues related to how a technology is used and by differentiating between personal factors and team or organizational factors. We are also taking a more thorough approach to measuring Perceived Usefulness, so the results do not simply show that usefulness leads to use (assuming the model is supported) but rather identify where Use Case Narratives are perceived to be most useful.

This research provides three main contributions. First, the results give us a better understanding of Use Case Narrative adoption and its role in the UML. Practitioner views on Use Case-driven development are largely unknown at this time, but we do know that this approach is far from universally favored. This understanding should contribute to the ongoing evolution of the UML and related methodologies. Second, the study extends the use of TAM to help understand how a technology is used rather than simply whether it is used. Third, this study provides a framework for future similar studies into other UML diagrams and their constructs. This includes providing validated measures of key constructs, including Use Case-driven, that could be useful outside of TAM as well. In addition, future studies might test for differences in factors affecting Perceived Usefulness depending on the job responsibilities of respondents or other demographic factors.

References

Adams, D. A., Nelson, R. R., and Todd, P. A. 1992. "Perceived Usefulness, Ease of Use, and Usage of Information Technology - A Replication," *MIS Quarterly* (16:2), pp. 227-247.

Agarwal, R., and Prasad, J. 1997. "The Role of Innovation Characteristics and Perceived Voluntariness in the Acceptance of Information Technologies," *Decision Sciences* (28:3), pp. 557-582.

Agarwal, R., and Prasad, J. 2000. "A field study of the adoption of software process innovations by information systems professionals," *IEEE Transactions on Engineering Management* (47:3), pp. 295-308.

Ajzen, I. 1988. *Attitudes, Personality and Behavior*, Buckingham, UK: Open University Press.

Ambler, S. W. 2002. *Agile Modeling,* New York, NY: John Wiley & Sons.

Anderson, D. 1999. "Are Use Cases the death of good UI Design?," *UI Design.Net* (available online at http://www.uidesign.net/1999/imho/feb_imho.html; accessed September 4, 2010.

Anderson, D. 1999. "Use Cases still considered Dangerous!," *UI Design.Net* (available online at http://www.uidesign.net/1999/imho/oct_imho.html; accessed September 4, 2010.

Arlow, J., and Neustadt, I. 2004. *Enterprise Patterns and MDA: Building Better Software with Archetype Patterns and UML*, Boston, MA: Addison-Wesley.

Armour, F., and Miller, G. 2001. *Advanced Use Case Modeling*, Boston, MA: Addison-Wesley.

Ashrafi, N., and Ashrafi, H. 2009. *Object-Oriented Systems Analysis and Design*, Upper Saddle River, NJ: Prentice Hall.

Bagozzi, R. P. 2007. "The Legacy of the Technology Acceptance Model and a Proposal for a Paradigm Shift," *Journal of the Association for Information Systems* (8:4), pp. 244-254.

Bahrami, A. 1999. *Object Oriented Systems Development,* Boston, MA: Irwin McGraw-Hill.

Booch, G., Rumbaugh, J., and Jacobson, I. 1999. *The Unified Modeling Language User Guide*, Reading, MA: Addison-Wesley.

Burton-Jones, A., and Hubona, G. S. 2005. "Individual Differences and Usage Behavior: Revisiting a Technology Acceptance Model Assumption," *Database for Advances in Information Systems* (36:2), pp. 58-101.

Burton-Jones, A., and Straub, D. W. 2006. "Reconceptualizing System Usage: An Approach and Empirical Test," *Information Systems Research* (17:3), pp. 228-246.

Chau, P. Y. K. 1996a. "An Empirical Assessment of a Modified Technology Acceptance Model," *Journal of Management Information Systems* (13:2), pp. 185-204.

Chau, P. Y. K. 1996b. "An empirical investigation on factors affecting the acceptance of CASE by systems developers," *Information & Management* (30:6), pp. 269-280.

Chau, P. Y. K., and Hu, P. J.-H. 2001. "Information technology acceptance by individual professionals: A model comparison approach," *Decision Sciences* (32:4), pp. 699-719.

Chau, P. Y. K., and Hu, P. J. 2002. "Examining a model of information technology acceptance by individual professionals: An exploratory study," *Journal of Management Information Systems* (18:4), pp. 191-229.

Chin, W. W. 1998. "Issues and opinions on structural equation modeling," *MIS Quarterly* (22:1), pp. vii-xvi.

Chin, W. W. 2010. "How to write up and report PLS analyses," in *Handbook on Partial Least Squares*, V. Esposito Vinzi, W.W. Chin, J. Henseler, and H. Wang, (eds.), Heidelberg, Germany: Springer Verlag, pp. 655–690.

Chin, W. W., and Gopal, A. 1995. "Adoption intention in GSS: Relative importance of beliefs," *DATABASE* (26**:**2/3), pp. 42–64.

Cockburn, A. 2001. *Writing Effective Use Cases,* Boston, MA: Addison-Wesley.

Davis, F. D., Jr. 1986. *Technology Acceptance Model for Empirically Testing New End-User Information Systems: Theory and Results*, Unpublished Ph.D. Dissertation, Massachusetts Institute of Technology, Boston, MA, 1986.

Davis, F. D. 1989. "Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology," *MIS Quarterly* (13:3), pp. 319-340.

Davis, F. D., Bagozzi, R. P., and Warshaw, P. R. 1989. "User Acceptance of Computer Technology: A Comparison of Two Theoretical Models," *Management Science* (35:8), pp. 982-1003.

Dennis, A., Haley Wixom, B., and Tegarden, D. 2005. *Systems Analysis and Design with UML Version 2.0: An Object-Oriented Approach,* 2nd ed., Hoboken, NJ: John Wiley & Sons.

Dishaw, M. T., and Strong, D. M. 1998. "Supporting software maintenance with software engineering tools: A computed task-technology fit analysis," *Journal of Systems and Software* (44:2), pp. 107-120.

Dishaw, M. T., and Strong, D. M. 1999. "Extending the Technology Acceptance Model with Task-technology Fit Constructs," *Information & Management* (36:1), pp. 9-21.

Dishaw, M. T., and Strong, D. M. 2003. "The effect of task and tool experience on maintenance CASE tool usage," *Information Resources Management Journal* (16:3), pp. 1-16.

Dobing, B., and Parsons, J. 2000. "Understanding the role of use cases in UML: A review and research agenda," *Journal of Database Management* (11:4), pp. 28-36.

Dobing, B., and Parsons, J. 2006. "How UML is used," *Communications of the ACM* (49:5), pp. 109-113.

Dobing, B., and Parsons, J. 2008. "Dimensions of UML Diagram Use: A Survey of Practitioners," *Journal of Database Management* (19:1), pp.1-18.

Dzidek, W. J., Arisholm, E., and Briand, L. C. 2008. "A Realistic Empirical Evaluation of the Costs and Benefits of UML in Software Maintenance," *IEEE Transactions on Software Engineering* (34:3), pp. 407-432.

Esposito Vinzi, V., Trinchera, L., and Amato, S. 2010. "PLS path modeling: From foundations to recent developments and open issues for model assessment and improvement," in *Handbook on Partial Least Squares*, V. Esposito Vinzi, W. W. Chin, J. Henseler, and H. Wang, (eds.), Heidelberg, Germany: Springer Verlag, pp. 47–81.

Fichman, R. G., and Kemerer, C. F. 1997."The assimilation of software process innovations: An organizational learning perspective," *Management Science* (43:10), pp. 1345-1363.

Fitzgerald, B. 1998. "An empirical investigation into the adoption of systems development methodologies," *Information & Management* (34:6), pp. 317-328.

Fornell, C., and Larcker, D. F. 1981. "Evaluating structural equation models with unobservable variables and measurement error," *Journal of Marketing Research* (18:1), pp. 39–50.

Fowler, M. 1998. *Use and Abuse Cases*, (available online at http://www.martinfowler.com/distributedComputing/abuse.pdf; accessed September 4, 2010).

Gefen, D., and Straub, D. 2005. "A practical guide to factorial validity using PLS-Graph: Tutorial and annotated example," *Communications of the Association for Information Systems* (16:5), pp. 91-105.

Gefen, D., Straub, D. W., and Boudreau, M.-C. 2000. "Structural equation modeling and regression: Guidelines for research practice," *Communications of the Association for Information Systems* (4:7), pp. 3-77.

Glinz, M. 2000. "Problems and Deficiencies of UML as a Requirements Specification Language," in *Proceedings of the Tenth International Workshop on Software Specification and Design*, San Diego, CA, pp. 11-22.

Goetz, O., Liehr-Gobbers, K., and Krafft, M. 2010. "Evaluation of structural equation models using the partial least squares (PLS) approach," in V. Esposito Vinzi, W.W. Chin, J. Henseler, and H. Wang, (eds.), *Handbook on Partial Least Squares*, Heidelberg, Germany: Springer Verlag, pp. 661–711.

Goodhue, D. L. 1998. "Development and measurement validity of a task-technology fit instrument for user evaluations of information systems," *Decision Sciences* (29:1), pp. 105-138.

Goodhue, D. L., and Thompson, R. L. 1995. "Task-technology fit and individual performance," *MIS Quarterly* (19:2), pp. 213-236.

Grossman, M., Aronson, J. E., and McCarthy, R. V. 2005. "Does UML make the grade? Insights from the software development community," *Information and Software Technology* (47:6), pp. 383-397.

Hardgrave, B. C. 1995. "When to prototype: Decision variables used in industry," *Information and Software Technology* (37:2), pp. 113-118.

Hardgrave, B. C., Davis, F. D., and Riemenschneider, C. K. 2003. "Investigating determinants of software developers' intentions to follow methodologies," *Journal of Management Information Systems* (20:1), pp. 123-151.

Hardgrave, B. C., and Johnson, R. A. 2003. "Toward an information systems development acceptance model: The case of object-oriented systems development," *IEEE Transactions on Engineering Management* (50:3), pp. 322-336.

Harmon, P., and Watson, M. *Understanding UML: The Developer's Guide*, San Francisco, CA: Morgan Kaufmann, 1998.

Honaker, J., King, G., and Blackwell, M. 2010. *Amelia II: A program for missing data*. Version 1.2-17, (available online at http://cran.r-project.org/web/packages/Amelia/Amelia.pdf; accessed September 4, 2010).

Huisman, M., and Iivari, J. 2006. "Deployment of systems development methodologies: Perceptual congruence between IS managers and systems developers," *Information & Management* (43:1), pp. 29-49.

Hulland, J. 1999. "Use of partial least squares (PLS) in strategic management research: A review of four recent studies," *Strategic Management Journal* (20:2), pp. 195–204.

Iivari, J. 1996. "Why are CASE tools not used?," *Communications of the ACM* (39:10), pp. 94-103.

Jacobson, I., Booch, G., and Rumbaugh, J. 1999. *The Unified Software Development Process,* Reading, MA: Addison, Wesley.

Jacobson, I., Christerson, M., Jonsson, P., and Overgaard, G. 1992. *Object-Oriented Software Engineering: A Use Case Driven Approach*, Workingham, England: Addison-Wesley.

King, W. R., and He, J. 2006. "A meta-analysis of the technology acceptance model," *Information & Management* (43:6), pp. 740-755.

Kobryn, C. 1999. "UML 2001: A Standardization Odyssey," *Communications of the ACM* (42:10), October pp. 29-37.

Langle, G. B., Leitheiser, R. L., and Naumann, J. D. 1984. "A Survey of Applications Systems Prototyping in Industry," *Information & Management* (7:5), pp. 273-284.

Larman, C. 1998. *Applying UML And Patterns: An Introduction to Object-Oriented Analysis and Design*, Upper Saddle River, NJ: Prentice Hall.

Lee, S. M., Kim, I., Rhee, S., and Trimi, S. 2006. "The role of exogenous factors in technology acceptance: The case of object-oriented technology," *Information & Management* (43:4), pp. 469-480.

Ludlow, L. 2007. "The Application of User Stories for Strategic Planning," in *Proceedings of the 8th International Conference on Agile Processes in Software Engineering and Extreme Programming,* Como, Italy, pp. 198-202.

Mathieson, K., Peacock, E., and Chin, W. W. 2001. "Extending the technology acceptance model: The influence of perceived user resources," *Database for Advances in Information Systems* (32:3), pp. 86-112.

Meyer, B. 1997. *Object-Oriented Software Construction*, 2nd ed., Upper Saddle River, NJ: Prentice Hall.

Mills, D. 2002. "What's the Use of a Use Case?," *Software Education* (available online at http://www.softed.com/Resources/Docs/UseCase.aspx; accessed September 4, 2010).

Moore, G. C., and Benbasat, I. 1991. "Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation," *Information Systems Research* (2:3), pp. 192-222.

Object Management Group. 2009a. *Unified Modeling Language (OMG UML) Superstructure Version 2.2*. (available online at http://www.omg.org/spec/UML/2.2/; accessed September 4, 2010).

Object Management Group. 2009b *Unified Modeling Language (OMG UML) Infrastructure Version 2.2*. (available online at http://www.omg.org/spec/UML/2.2/; accessed September 4, 2010).

Pender, T. 2003. *UML Bible,* Indianapolis, IN: Wiley Publishing.

Periyasamy, K., and Ghode, A. 2009. "Cost Estimation using extended Use Case Point (e-UCP) Model," in *Proceedings of the International Conference on Computational Intelligence and Software Engineering,* Wuhan, China, pp. 1-5.

Pijpers, G. G. M., and van Montfort, K. 2006. "An Investigation of Factors that Influence Senior Executives to Accept Innovations in Information Technology," *International Journal of Management* (23:1), pp. 11-23.

Ratanotayanon, S., Sim, S. E., and Gallardo-Valencia, R. 2009. "Supporting Program Comprehension in Agile with Links to User Stories," in *Proceedings of the Agile 2009 Conference,* Chicago, IL.

Riemenschneider, C. K., and Hardgrave, B. C. 2001. "Explaining software development tool use with the technology acceptance model," *Journal of Computer Information Systems* (41:4), pp. 1-8.

Riemenschneider, C. K., Hardgrave, B. C., and Davis, F. D. 2002. "Explaining software developer acceptance of methodologies: A comparison of five theoretical models," *IEEE Transactions on Software Engineering* (28:12), pp. 1135-1145.

Roberts, T. L., Gibson, M. L., and Fields, K. T. 1999. "System development methodology implementation: Perceived aspects of importance," *Information Resources Management Journal* (12:3), pp. 27-38.

Roberts, T. L., Jr., Gibson, M. L., Fields, K. T., and Rainer, R. K., Jr. 1998. "Factors that impact implementing a system development methodology," *IEEE Transactions on Software Engineering* (24:8), pp. 640-649.

Rogers, E. M. 1983. *Diffusion of innovations,* 3rd ed., New York, NY: Free Press.

Rosenberg, D., and Scott, K. 2001. *Applying Use Case Driven Object Modeling with UML: An Annotated E-Commerce Example*, Boston, MA: Addison-Wesley.

Rumbaugh, J., Jacobson, I., and Booch, G. 2005 *The Unified Modeling Language Reference Manual*, (2nd ed.) Boston, MA: Addison-Wesley.

Sanchez, G. and Trinchera, L. 2009. plspm: Partial Least Squares data analysis methods. R package version 0.1-11. (available at http://cran.r-project.org/web/packages/plspm/plspm.pdf; accessed September 4, 2010).

Seacord, R. C., Plakosh, D., and Lewis, G. A. 2003. *Modernizing Legacy Systems: Software Technologies, Engineering Process and Business Practices,* Boston, MA, Pearson Education.

Siau, K., and Cao, Q. 2001. "Unified Modeling Language (UML) - a complexity analysis," *Journal of Database Management* (12:1), pp. 26-34.

Straub, D., Boudreau, M.-C., and Gefen, D. 2004. "Validation guidelines for IS positivist research," *Communications of the Association for Information Systems* (13:24), pp. 380-427.

Strong, D. M., Dishaw, M. T., and Bandy, D. B. 2006. "Extending Task Technology Fit with Computer Self-Efficacy," *Database for Advances in Information Systems* (37:2/3), pp. 96-106.

Tenenhaus, M., Amato, S., and Esposito Vinzi, V. 2004. "A global goodness-of-fit index for PLS structural equation modeling," in *Atta della XLII Riunione Scientifica,* Bari, Italy, 9–11, pp. 739-742.

Thompson, R., Compeau, D., and Higgins, C. 2006. "Intentions to Use Information Technologies: An Integrative Model," *Journal of Organizational and End User Computing* (18:3), pp. 25-46.

Thompson, R. L., Higgins, C. A., and Howell, J. M. 1991, "Personal Computing: Toward a Conceptual Model of Utilization," *MIS Quarterly* (15:1), pp. 125-143.

Venkatesh, V., and Davis, F. D. 1996. "A Model of the Antecedents of Perceived Ease of Use: Development and Test," *Decision Sciences* (27:3), pp. 451-481.

Venkatesh, V., and Davis, F. D. 2000. "A theoretical extension of the technology acceptance model: Four longitudinal field studies," *Management Science* (46:2), pp. 186-204.

Venkatesh, V., Speier, C., and Morris, M. G. 2002. "User acceptance enablers in individual decision making about technology: Toward an integrated model," *Decision Sciences* (33:2), pp. 297-316.

Zeichick, A. 2002. "Modeling Usage Low; Developers Confused About UML 2.0, MDA," *SD Times*, July 15, (available online at http://www.sdtimes.com/article/story-20020715-03.html; accessed September 4, 2010).

Zeichick, A. 2004. "UML Adoption Making Strong Progress," *SD Times*, August 15, (available online at http://www.sdtimes.com/article/story-20040815-14.html; accessed September 4, 2010).

Appendix: Questionnaire Items

Attitude

Att1	I believe that Use Case Narratives should be written when the UML is used to develop systems.
Att2	I believe Use Case Narratives play an important role when developing systems using the UML.
Att3	I believe that Use Case Narratives should be used on most UML projects.

Intentions for Others to Use

Int.Others1	Given the opportunity, I would strongly encourage my team to write Use Case Narratives when using the UML on future system development projects.
Int.Others2	If anyone asks my opinion, I would strongly encourage them to employ Use Case Narratives on UML projects.

Intentions for Self to Use

Int.Self1	I would employ Use Case Narratives on projects whenever possible.
Int.Self2	Given the opportunity, I would employ Use Case Narratives on my future system development projects.

Perceived Usefulness

Use1	Use Case Narratives can be very effective in UML projects.
Use2	Use Case Narratives can play an important role in UML projects.
Use3	Use Case Narratives can be very useful in UML projects.

Usefulness for Communication

Comm1	Use Case Narratives are very helpful in ensuring that system analysts understand what the clients and users have in mind.
Comm2	Use Case Narratives are very helpful in ensuring that the clients and users understand what system analysts are proposing.
Comm3	Use Case Narratives help avoid the miscommunication problems that can occur between clients/users and system analysts.

Usefulness for Programmers

Prog1	Programmers find Use Case Narratives very helpful.
Prog2	Programming is easier when programmers have read the relevant Use Case Narratives.
Prog3	Use Case Narratives are useful for programmers.

Usefulness for Project Management

Manage1	Writing Use Case Narratives helps to estimate the size of the system to be developed.
Manage2	Writing Use Case Narratives helps to estimate the time and effort required for the development of a planned system.
Manage3	Writing Use Case Narratives helps to prioritize the development of different parts of the system.
Manage4	Use Case Narratives help us deliver systems on schedule and within budget.

Usefulness for Efficiency

Eff1	Writing Use Case Narratives enables us to build systems more quickly.
Eff2	Writing Use Case Narratives enables us to increase our productivity when building systems.
Eff3	Writing Use Case Narratives makes it easier to build systems.

Usefulness for Maintenance	
Maint1	Reviewing Use Case Narratives helps when adding new functionality to existing systems.
Maint2	Reviewing and updating Use Case Narratives helps to decrease the cost of enhancements to the system.
Maint3	Reviewing and updating Use Case Narratives allows us to better manage enhancements to the system.
Usefulness for System Quality	
Qual1	Writing Use Case Narratives helps ensure that the final system meets the functional requirements.
Qual2	Writing Use Case Narratives results in higher quality systems.
Qual3	Systems that are developed based on Use Case Narratives generally have higher quality.
Usefulness for Developing Class Diagrams	
ClassD1	Use Case Narratives are very helpful when creating Class Diagrams and/or Data Models.
ClassD2	Use Case Narratives are helpful in identifying the Classes (or Entities) we need for a Class Diagram and/or Data Model.
ClassD3	Use Case Narratives are helpful in determining the Attributes needed for Classes (or Entities) in a Class Diagram and/or Data Model.
ClassD4	Use Case Narratives are helpful in determining the Methods needed for Classes in a Class Diagram.
Usefulness for User Interface Design	
UI1	Use Case Narratives are very useful when designing the user interface.
UI2	The user interface should be designed based on Use Case Narratives.
UI3	Having Use Case Narratives makes the user interface design task much easier.
Usefulness for Testing	
Test1	Use Case Narratives are very useful when testing systems.
Test2	Use Case Narratives provide a good framework for system testing.
Test3	Use Case Narratives are valuable for system testing.
Ease of Use	
EoU1	I believe that creating and organizing Use Case Narratives is straightforward.
EoU2	Creating Use Case Narratives is not difficult.
EoU3	Creating Use Case Narratives is something most system analysts should be able to do.
EoU4	Creating Use Case Narratives can be done efficiently.

Note: UI2 and EoU3 are not included in the analysis.