Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2006 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2006

Diffusion of Model-Driven Architecture in Academia

Thomas Meservy
University of Arizona

Kelly Fadel
University of Arizona

Kurt Fenstermacher
University of Arizona

Jon Marthaler
University of Arizona

Diffusion of Model-Driven Architecture in Academia

Thomas O. Meservy
University of Arizona
tmeservy@gmail.com

Kelly J. Fadel
University of Arizona
kjf@eller.arizona.edu

Kurt D. Fenstermacher
University of Arizona
kurtf@eller.arizona.edu

Jon Marthaler
University of Arizona
jmarthaler@gmail.com

ABSTRACT

Model-driven architecture (MDA) is an innovative software development paradigm that advocates the use of modeling throughout the software development lifecycle. This paradigm appears to be gaining acceptance in industry as several large companies are investing resources to develop MDA-compliant tools. However, we know little about the diffusion of MDA within academia, specifically in the teaching of systems analysis and design. A survey of information systems and computer science faculty from around the world was conducted to understand the adoption lifecycle; initial analysis indicates that academia is early in the adoption lifecycle. We also consider the correlation of individuals' perceived benefits of MDA and each stage of the adoption lifecycle.

Keywords

Innovation Diffusion, Model-Driven Architecture, Software Analysis and Design, IS Education

INTRODUCTION

Information systems (IS) educators face the daunting challenge of maintaining a curriculum that tracks with changes in technology and business needs (Davis, Gorgone, Couger, Feinstein and Longenecker, 1997; Mawhinney, Morrell, Morris and Helms, 1995; Mawhinney, Morrell, Morris and Munro, 1999; Noll and Wilkins, 2002). Both the information technology (IT) industry and academia have an incentive to ensure that graduates will be well versed in the latest technologies and methods (Gallivan, Truex and Kvansky, 2002; Noll et al., 2002). Understanding the diffusion of a new technology in an academic setting offers a better understanding of how well the IS curriculum tracks industry trends, awareness of a new technology, perceived industry demand for the new technology, the perceived benefits of the new technology, and the importance of those benefits.

Systems analysis and design is a domain of information systems with both high industry demand and rapid evolution of technologies and practices (Misic and Russo, 1999). A relatively new innovation is the Object Management Group's Model-Driven Architecture (MDA), which recasts the software development process as the creation of models, which MDA-compliant tools then transform into successively more concrete models and eventually into code. When the system requirements change, the models are updated and the code is re-generated, ensuring that the models are always synchronized with the code. Most importantly, MDA changes the primary artifact of development from code to models (Object Management Group, 2001). In this paper, we investigate the diffusion of MDA in academia through the use of a repeated survey distributed to both computer science and information systems educators. The survey had two purposes: 1) to assess the state of diffusion of MDA as a teaching tool in academic institutions, and 2) to investigate the relationships among the perceived benefits of MDA, adopter characteristics, and the observed pattern of diffusion. The results of the survey illustrate that the diffusion of MDA in academia is in its early stages.

BACKGROUND

Model-Driven Architecture

MDA, published by the Object Management Group (OMG) in 2001, is a software development paradigm that advocates the use of models throughout the software development lifecycle. (While the OMG has advanced MDA, researchers have studied the broader notion of model-driven development for some time. We focused specifically on MDA because of its industry backing and the specificity of the concept.) MDA advocates separating the problem domain from the problem-solving technology and suggests seeing problems from successively more detailed viewpoints; each viewpoint has a corresponding model. (Meservy and Fenstermacher (2005) offer a brief tutorial on MDA and a discussion of how it changes the focus of the software development process.) At the most abstract level, the model focuses on the problem domain and is given in terms familiar to domain experts, rather than technologists; the computation-independent model (CIM) is a business model that is divorced from technology. At the next level, the platform-independent model (PIM) does not commit to a particular technology, but instead simply changes the perspective from domain expert to technologist. At some point, the developers will choose a specific platform, for example, Sun's Java 2 Enterprise Edition, and transform the PIM into a Platform-Specific Model (PSM). MDA uses automated transformation patterns to move between abstraction layers, and the automation supports regeneration of other models when one changes. The PSM is eventually transformed into executable code and deployed in a specific environment.

MDA has several potential benefits, including reduced risk of project failure due to changing technology (Brassard and Cardinal, 2002), improved productivity for developers and architects (Brassard et al., 2002), business models and technologies that can evolve independently (Brassard et al., 2002), simpler enforcement of best practices, standards, guidelines (Sirosky, 2003), and application quality and reliability (Sirosky, 2003). Despite these benefits, some have cited potential problems with MDA, including its lack of maturity, steep learning curve, and high upfront cost (Weiszmann and Messenheimer, 2005). These issues, according to critics, may relegate MDA to becoming the next disappointment in a long line of highly touted "silver bullets" in software development. Nevertheless, a growing number of MDA success stories (www.omg.org/mda/products_success.htm) and its increasing vendor support challenge this prediction, and indicate that MDA will continue its trend toward becoming a dominant software development paradigm.

Diffusion of Innovation

The diffusion of innovation has been widely studied over the years (Fichman, 2000; Larsen and McGuire, 1998). Hagerstrand (1953) asserted that the adoption process is primarily the outcome of a learning and communication process. Kwon and Zmud (1987) focus on a change agent that creates a new technology and then directs its diffusion within a particular social system. Rogers (1995) describes diffusion as a process in which an innovation is communicated through certain channels over time among members of a specific social system. The diffusion of an innovation in most social systems is not a collective decision; rather, each individual makes her or his individual decision of adoption. Thus, adoption of an innovation can be viewed as an individual decision and diffusion of an innovation is a combined, snapshot view of the social system's adoption.

Rogers (1995) breaks down the process of innovation adoption into five, main progressive stages: awareness, interest, evaluation, trial, and adoption. Awareness is knowing that an innovation exists, but not having a working knowledge of the innovation. Interest is when an individual seeks information about the innovation. Evaluation means mentally applying the innovation to a situation and deciding whether to try it. Trial means using the innovation in the short-term. Finally, adoption is continuing the full use of the innovation.

In addition to considering each individual's stage of adoption, researchers categorize adopters by when they adopt a particular innovation (relative to the whole social system). Major groups ordered chronologically by adoption date include innovators (first 2.5% of adopters), early adopters (next 13.5% of adopters), early majority (next 34% of adopters), late majority (next 34% of adopters), and laggards (last 16% of adopters); this distribution produces the familiar S-shaped curve for cumulative adoption (Rogers, 1995).

METHOD

We conducted two surveys of information systems and computer science faculty to understand the diffusion of MDA in an academic setting and why certain individuals adopted earlier than others. We also included questions about the perceived benefits of MDA to determine if adoption rates correlated with perceived benefits. We surveyed faculty twice, a year apart, to measure changes in adoption. Participants were recruited from the IS World mailing list and the computer science educator's mailing list, Both surveys included demographic questions as well as questions to gauge the awareness of MDA by the survey takers. The demographics section contained questions that dealt with the background of the individual and the organization where they work. Information was gathered about the type of academic appointment, the importance of teaching and research at the institution, whether the department has a masters or doctoral program, what college the department belongs to, and what classes the individual has taught and is slated to teach.

The survey included several groups of questions, organized by stage of adoption, to assess an individual's adoption of MDA. Questions in the awareness section asked whether the individual had heard of MDA (prior to the survey) and the extent of her knowledge. The interest section contained questions that included a self-ranking of the individual's interest in MDA and the time spent studying MDA. We used the amount of time that the individual had spent reading or researching MDA to assess the evaluation stage. This section also queried the perceived impact of MDA on various aspects of software development and the importance of those aspects. The trial section of the survey had questions asking about the amount of time spent evaluating MDA products, awareness of specific MDA products, whether the individual had used any MDA products, and which products had been used. The adoption section had questions that focused on the current or planned use of MDA in an educational setting. Sample questions included whether MDA had been or would be taught in a classroom setting and which products were used. There were additional questions about the classes where MDA would be taught, including the type of class, how much time would be devoted to MDA, and the amount of emphasis that would be placed on various development concepts.

RESULTS

Demographics

Of the 39 respondents that participated in the first survey, 21 were from a computer science educators' mailing list (SIGCSE, http://www.sigcse.org/join/list.shtml) and 18 were from the Association for Information Systems IS World mailing list (http://lyris.isworld.org/isworldlist.htm). Of all the respondents, 41% were tenured, 36% were tenure-track but not tenured, and the remaining were either not tenure-track or graduate students. Just over 60% indicated that teaching was very important at their institution; 46.2% marked that research was very important at their institution.

A total of 18 participants, 7 from the IS World list and 11 first-survey participants, responded to the second survey. In the second survey, 56% were tenured, 17% were tenure-track but not tenured, and the remaining were either not tenure-track or graduate students. Of those who responded, 56% indicated that teaching was very important at their institution and 61% marked that research was very important at their institution. Of those who took both surveys, 63.6% were initially recruited from the IS World list and 36.4% were recruited from the CS educators mailing list and 54.5% were tenured. Interestingly, only 36.4% of this group indicated that teaching was very important and 54.5% reported research as being very important at their institutions.

Diffusion Stage

As indicated previously, questions in the survey were written to assess the individual's current stage in the adoption process. Figure 1 illustrates the percent of people who responded positively to each of the questions listed. For example, 66.7% of individuals had heard of MDA previous to the study.

We assessed each individual's adoption stage by combining responses to multiple questions. Figure 2 displays both cumulative and incremental percentages for each adoption stage. We considered an individual to be in or have passed through the awareness stage if she had heard of MDA prior to the survey and had at least minimal knowledge of it (74.4% of respondents are either in the awareness stage or have passed through it to a later stage). Those that responded as having interest in MDA and reported reading or researching MDA at least a little were categorized as being in or having passed through the interest stage (71.8%). If an individual had read or researched more than 20 hours or she had spent some time evaluating MDA products we considered her as in or having passed through the the evaluation stage because she had likely tried to understand how she might use or teach it (48.7%). If an individual reported time evaluating MDA products and specified that she had used a specific MDA product (e.g. OptimalJ, AndroMDA, other) then she was classified as being in the trial or adoption stages (28.2%). We considered those who had taught MDA in the past and planned to teach it in the future as being in the adoption stage (23.1%). Given the current adoption rate of 23%, we would characterize academics who adopt MDA in the near future as part of the early majority.

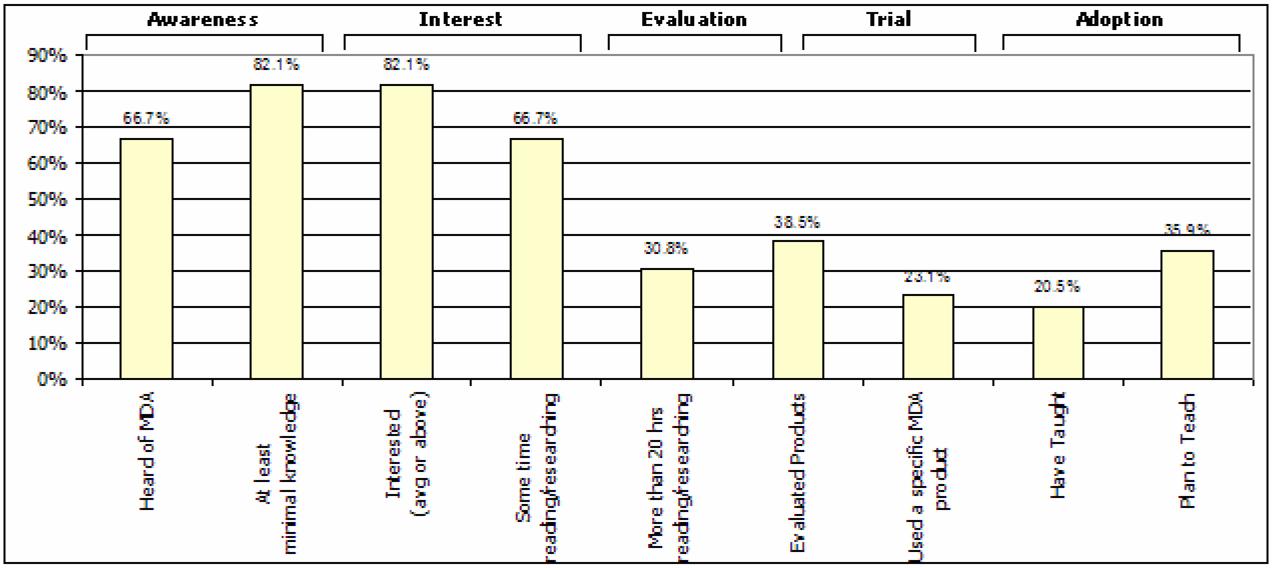

Figure 1. Responses to 2004 Survey Questions

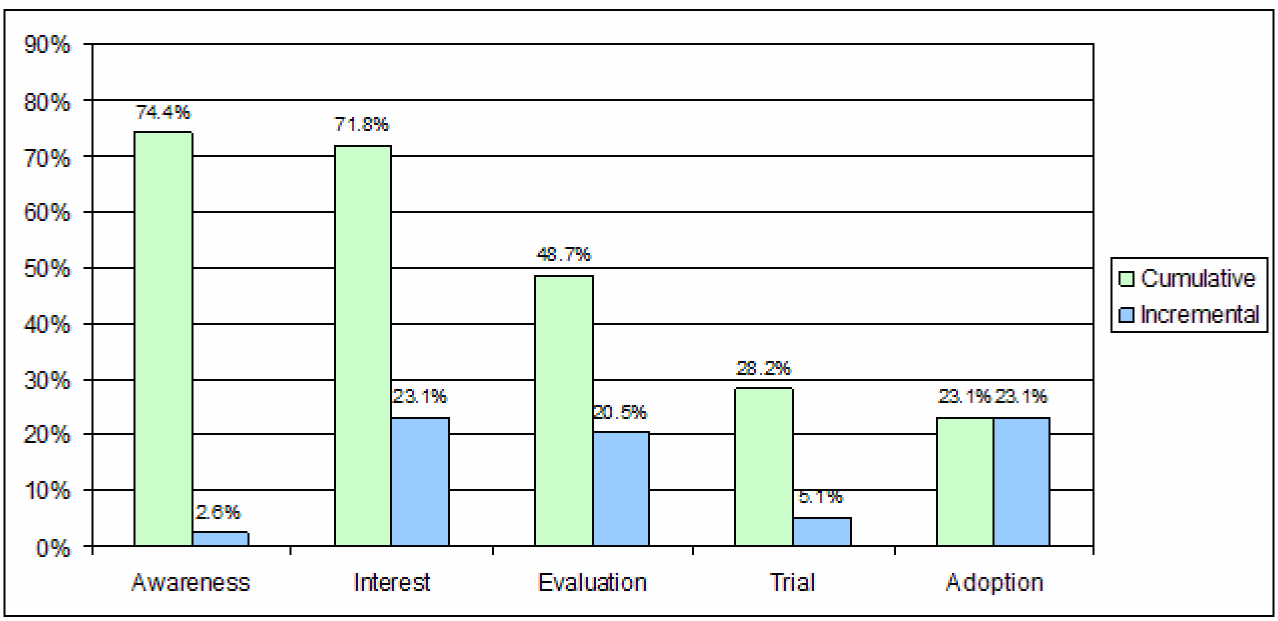

Figure 2. 2004 Survey Stages of Adoption

As mentioned previously, a second survey was conducted approximately one year after the first to reassess the adoption and use of MDA in academia. We used identical criteria to assess the current stage of the MDA adoption lifecycle for each participant. Figure 3 displays summarized responses for the 2005 survey and Figure 4 displays the cumulative and incremental percentage of participants for each stage.

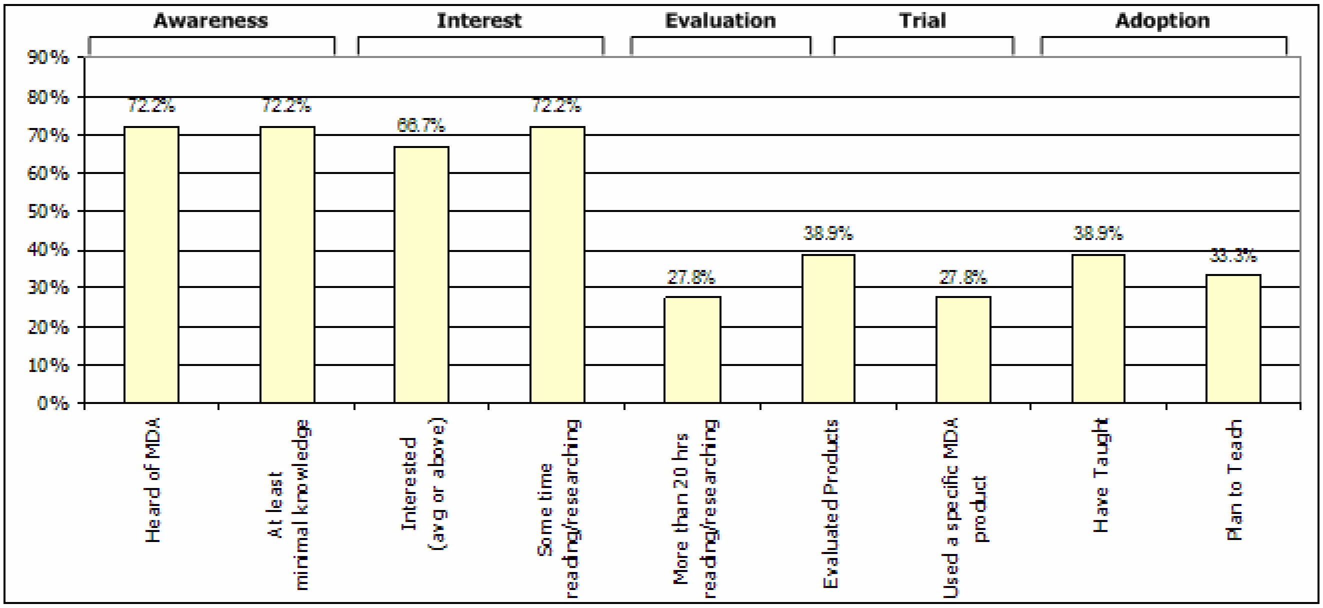

Figure 3. Responses to 2005 Survey Questions

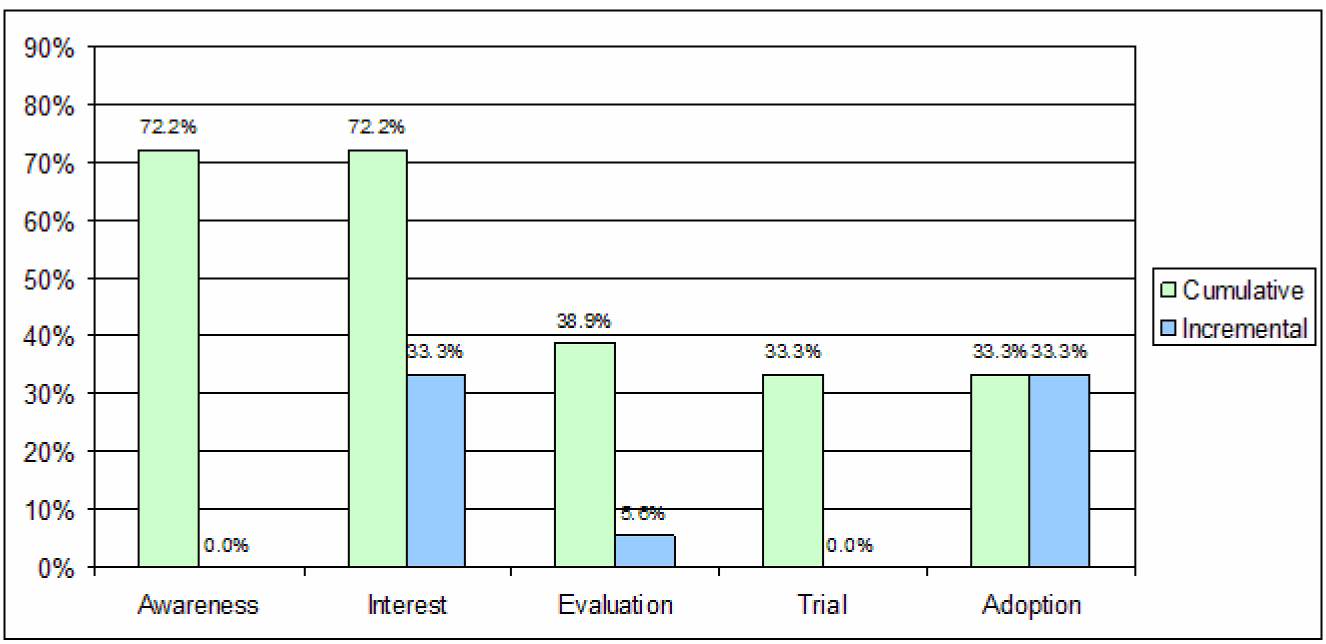

Figure 4. 2005 Survey Stages of Adoption

While overall awareness (72.2%) and interest (72.2%) remained approximately the same as the previous year, curiously, the number of people that responded as having evaluated MDA products or researched MDA more than 20 hrs dropped to 38.8%. As expected though, more people had tried MDA (33.3%) and adopted it (33.3%) for use in their teaching, though part of the increase may be attributable to participants that have invested time in MDA and also voluntarily participate in multiple surveys.

In addition to comparing responses from the entire participant pool between years, we wanted to see if any of the participants from the 2004 survey had moved further in the adoption lifecycle in 2005. Of the 11 individuals who indicated having participated previously, two were discarded. One was lacking identifying information to pair it with the previous response and another's answers were inconsistent between years (e.g. they indicated having evaluated, tried, and adopted MDA in 2004 but not in 2005). Figure 5 illustrates the responses for those who took the survey both years.

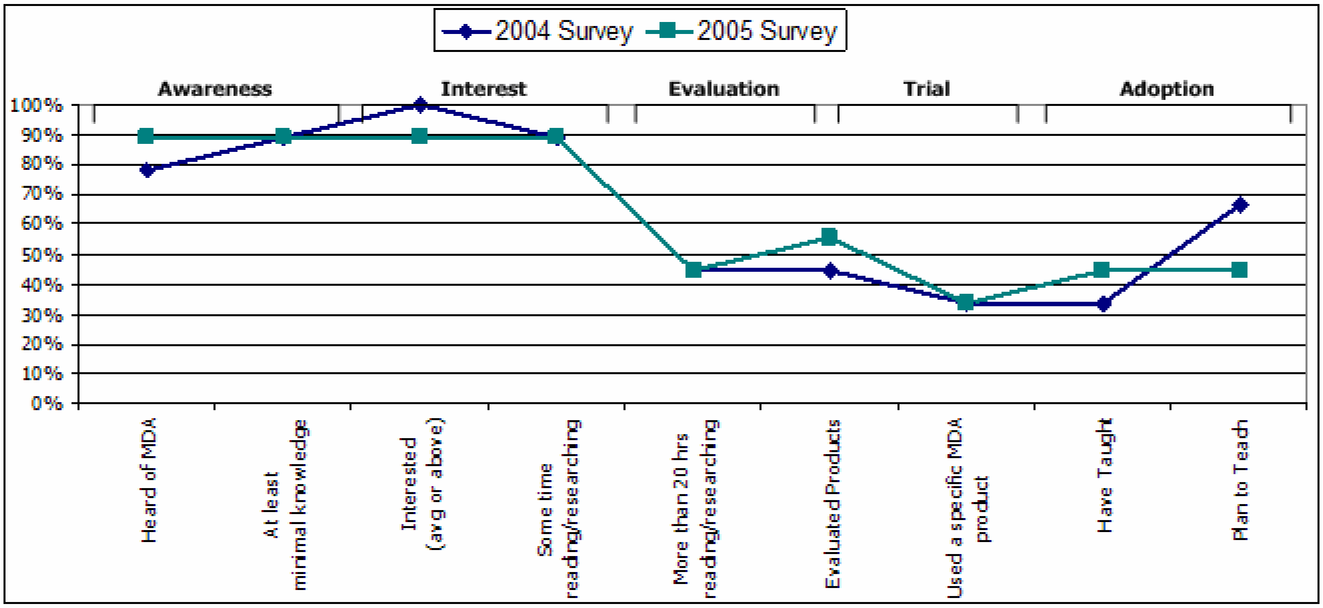

Figure 5. Responses to 2004 and 2005 Survey Questions For Participants of Both Surveys

For those who took the survey both years, the responses to questions were relatively stable across years and so were the proportion of individuals in each stage. In both years, the cumulative percentage for the first three stages of adoption were identical for this group of participants; awareness (88.8%), interest (88.8%), and evaluation (55.5%). Between 2004 and 2005 one additional individual progressed through the trial stage to the adoption stage increasing the percentage of individuals at each stage from 33% to 44%, although those who participated both years were further along in the adoption lifecycle than those who participated just once. Thus, while the responses of these individuals remained fairly constant, there was a slight shift toward increased use and teaching of MDA.

Even with the higher adoption rate suggested by the responses of all of the participants of the 2005 survey, we still categorize academics who adopt MDA in the near as the early majority (16.1%-54%); moreover, we expect increased use of MDA in academia due to its perceived benefits both as a tool for development and also as a tool to help teach modeling. In the next section, we discuss the correlation between perceive benefits and academic MDA adoption.

PERCEIVED BENEFITS OF MDA AND ADOPTER CHARACTERISTICS

Perceived Benefits of MDA

Diffusion of Innovations theory identifies five characteristics that influence the adoption of an innovation (Rogers, 1995): relative advantage (compared to alternative technologies), compatibility (with existing practice), complexity, trialability (how easy is it to evaluate the innovation), and observability (as Rogers (1995) defines it, "the degree to which the results of an innovation are visible to others.") How we rate MDA in terms of these five characteristics depends strongly on the context: academia or industry. The cost of change is lower for academics than for many in industry, who often must change long-standing business practices and deeply rooted habits of developers; thus, a priori, we would expect diffusion to be more rapid in academia than in industry. In addition, often no single characteristic is compelling enough to ensure adoption; instead, a constellation of perceived benefits leads to adoption. We asked survey participants to rate the importance of specific software development characteristics and to rate MDA as an innovation on these characteristics; Table 1 captures these responses for each stage of the adoption lifecycle. The percentages listed in left-hand side of each column Table 1 (labeled ‘Import’) are the percentage of individuals who marked that a specific characteristic was ‘Important’ or ‘Very important’. The percentages listed on the right-hand side of each column (labeled ‘Impact’) are the percentage of those who viewed MDA as having a ‘Positive impact’ or a ‘Significant positive impact’. In both cases, the percentages are of those who provided ratings. The bottom row of Table 1 gives the total percentage of responses for importance and impact across all characteristics (rows), for each stage. Table 1 includes data from 47 unique individuals; for those respondents who participated in both the first and second round surveys, only second round responses were included.

Category	Benefit	Unaware		Awareness		Interest		Evaluation		Trial		Adoption		Overall	
		Import	Impact	Import	Impact	Import	Impact	Import	Impact	Import	Impact	Import	Impact	Import	Impact
Productivity	Lower cost of application development and management	100%	100%	100%	100%	73%	67%	86%	83%	50%	100%	82%	73%	77%	81%
	Reduced application time to market	75%	100%	100%	100%	55%	33%	86%	83%	50%	100%	82%	82%	73%	72%
	Increased return on technology investments	50%	100%	100%	100%	70%	67%	86%	60%	50%	100%	82%	55%	64%	74%
	Improved productivity for developers	75%	0%	100%	100%	80%	33%	86%	67%	50%	100%	91%	82%	65%	83%
	Improved productivity for architects	50%	0%	100%	0%	50%	67%	86%	83%	50%	0%	82%	55%	58%	70%
	Reduced time for code reviews	50%	100%	0%	0%	33%	17%	29%	60%	50%	100%	64%	64%	52%	44%
	Reduced risk of project failure	100%	100%	100%	0%	82%	80%	86%	60%	0%	100%	82%	55%	63%	81%
Software Attribute	Enhanced portability between platforms	75%	100%	100%	0%	73%	80%	43%	60%	50%	100%	82%	73%	71%	69%
	Enhanced interoperability between platforms	100%	100%	100%	100%	91%	80%	86%	60%	50%	100%	82%	45%	63%	86%
	Application quality	100%	0%	100%	100%	91%	67%	100%	40%	50%	100%	100%	73%	64%	94%
	Application reliability	100%	100%	100%	100%	91%	50%	71%	33%	50%	100%	90%	55%	54%	86%
Maintenance and Documentation	Rapid inclusion of emerging technology benefits into existing systems	50%	100%	100%	0%	36%	60%	43%	60%	50%	100%	45%	64%	63%	44%
	Rapid inclusion of best practices, standards, guidelines	100%	100%	0%	0%	64%	60%	57%	67%	50%	100%	91%	73%	68%	72%
	Business models and technologies that evolve at their own pace	75%	100%	0%	0%	60%	67%	67%	60%	50%	100%	73%	64%	64%	65%
Number of participants in each stage		14		1		12		7		2		11		47	
Total percentage of responses in each stage		7%	29%	100%	100%	47%	87%	78%	99%	50%	100%	100%	99%	53%	75%

Table 1. Importance of software development goals and the impact of MDA on those goals by adoption stage

Considering the totals by stage (shown in the bottom row of Table 1), those who are further along in the adoption lifecycle provided more responses, not only about the impact of MDA on the software characteristics but also about the importance of those characteristics. Even though the difference in means of a particular characteristic between stages might not be statistically significant (likely due to the small number of participants in each stage), we can still note a few general trends about the perceived benefit of MDA. Figure 6 illustrates a subset of these trends for the interest, evaluation, and adoption stages; the remaining stages had too few responses to draw any conclusions.

Before the distribution of the survey, it was assumed that an individual progressing through the MDA adoption lifecycle would gain knowledge about MDA and recognize its purported benefits. We saw this general trend for three of the characteristics; *reduced time for code reviews, reduced application time to market*, and *improved productivity for developers*. For example, 82% of raters in the adoption stage indicated that MDA would have a positive or significantly positive impact on the *improved productivity for developers* characteristics. In other cases we saw an unexpected declining trend including the characteristics of *increased return on technology investments*, *reduced risk of project failure*, and *enhanced interoperability between platforms*. Perhaps individuals who were in an earlier adoption stage bought into the “marketing hype” surrounding MDA but changed their perceptions as they gained more knowledge and experience with current MDA tools. Another plausible explanation for the decline could be that those in later stages of adoption likely have greater knowledge of the software development process in general and may view those specific characteristics as “pipe dreams.” In the case of *enhanced interoperability between platforms*, many existing MDA tools currently target a single platform and perhaps those who have evaluated these tools attribute the limitations of the tool to MDA itself. Many of the other software characteristics didn’t display any visible trends across stages of the adoption lifecycle.

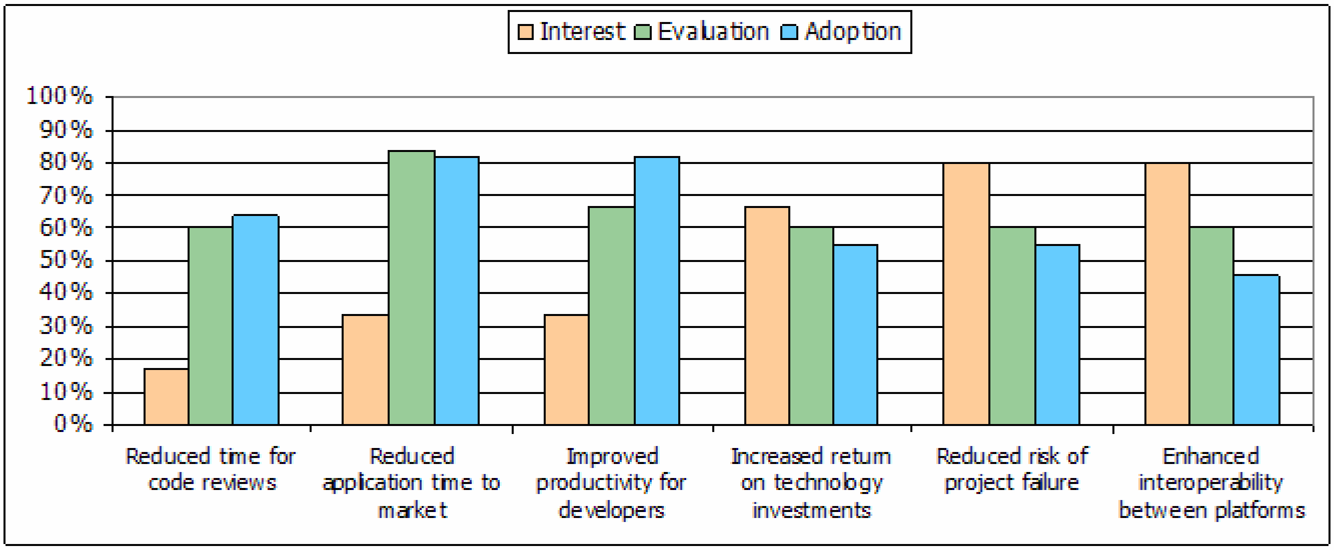

Figure 6. Percentage of individuals rating MDA as having a positive or significant positive impact on select software characteristics broken down by stage

We also correlated the stage of MDA (assuming a progression from unaware through adoption) with the perceived benefits of MDA using Spearman's ρ in a bivariate correlation analysis. Only *improved productivity for developers* ($\rho = 0.363$; $p = 0.038$) and *reduced time for code reviews* ($\rho = 0.357$; $p = 0.048$) were positively correlated at $\alpha = 0.05$. In other words, the further an individual is in the adoption lifecycle, the more highly she rated the impact of MDA on these categories.

Adopter Characteristics

Diffusion of Innovations theory also suggests that characteristics of individuals may influence their adoption patterns (Rogers, 1995). In the survey, we captured several demographic variables of respondents, including the type of position they held (tenure vs. non-tenure), the department they belonged to (IS vs. CS), the importance of teaching and research at their institution, the types of graduate programs offered at their institution (doctoral or masters programs), and the types of classes they taught or are teaching. Table 2 shows the demographic profiles for each of the five adoption stages. For example, of respondents categorized in the adoption stage (the far-right column), 36% belonged to IS departments and 64% belonged to CS departments. Table 3 shows the adoption profiles for each demographic variable measured. For example, of all respondents in IS departments (the first row of Table 3), 31% were in the unaware stage, 4% were in the awareness stage, 23% were in the interest stage, 19% were in the evaluation stage, 8% were in the trial stage, and 15% were in the adoption stage. Percentages in each of the tables are based on a total 47 responses (for respondents who participated in both the first and second round surveys, only second round responses were tabulated).

Variable	Value	Adoption Stage					
		Unaware	Awareness	Interest	Evaluation	Trial	Adoption
Department	IS	57%	100%	50%	71%	100%	36%
	CS	43%	0%	50%	29%	0%	64%
Appointment	Tenured	36%	100%	50%	29%	0%	64%
	Not Yet Tenured	29%	0%	17%	29%	50%	27%
	Non-Tenure Track	7%	0%	25%	14%	0%	0%
	Graduate Student	21%	0%	0%	14%	50%	0%
	Other	0%	0%	8%	14%	0%	9%
Degree Programs	Masters	64%	0%	75%	86%	50%	82%
	Doctoral	50%	100%	67%	71%	50%	64%
Type of Institution	Research Emphasis (Important or Very Important)	43%	100%	67%	86%	100%	45%
	Teaching Emphasis (Important or Very Important)	86%	100%	67%	71%	100%	55%
Courses Taught	Beginning Programming	57%	0%	67%	71%	50%	55%
	Intermediate Programming	36%	0%	67%	57%	50%	36%
	Advanced Programming	36%	0%	33%	14%	0%	27%
	Systems Analysis and Design (No Programming)	21%	100%	58%	71%	50%	64%
	Systems Analysis and Design (With Programming)	14%	100%	33%	57%	0%	64%
	Software Project Management	36%	0%	8%	0%	0%	27%
	Database Management Systems	36%	0%	58%	57%	50%	45%
	Web Development (with Server Side Programming)	14%	0%	8%	14%	0%	27%
	Web Development (without Server Side Programming)	29%	0%	8%	29%	0%	18%
Total Percentage of Responses in each Stage		30%	2%	26%	15%	4%	23%

Table 2. Demographic Profiles of Adoption Stages

Variable	Value	Adoption Stage						Total Percentage of Responses in each Category
		Unaware	Awareness	Interest	Evaluation	Trial	Adoption	
Department	IS	31%	4%	23%	19%	8%	15%	55%
	CS	29%	0%	29%	10%	0%	33%	45%
Appointment	Tenured	24%	5%	29%	10%	0%	33%	45%
	Not Yet Tenured	33%	0%	17%	17%	8%	25%	26%
	Non-Tenure Track	20%	0%	60%	20%	0%	0%	11%
	Graduate Student	60%	0%	0%	20%	20%	0%	11%
	Other	25%	0%	25%	25%	0%	25%	9%
Degree Programs	Masters	26%	0%	26%	18%	3%	26%	72%
	Doctoral	24%	3%	28%	17%	3%	24%	62%
Type of Institution	Research Emphasis (Important or Very Important)	21%	4%	29%	21%	7%	18%	60%
	Teaching Emphasis (Important or Very Important)	35%	3%	24%	15%	6%	18%	72%
Courses Taught	Beginning Programming	29%	0%	29%	18%	4%	21%	60%
	Intermediate Programming	23%	0%	36%	18%	5%	18%	47%
	Advanced Programming	38%	0%	31%	8%	0%	23%	28%
	Systems Analysis and Design (No Programming)	13%	4%	29%	21%	4%	29%	51%
	Systems Analysis and Design (With Programming)	11%	6%	22%	22%	0%	39%	38%
	Software Project Management	56%	0%	11%	0%	0%	33%	19%
	Database Management Systems	23%	0%	32%	18%	5%	23%	47%
	Web Development (with Server Side Programming)	29%	0%	14%	14%	0%	43%	15%
	Web Development (without Server Side Programming)	44%	0%	11%	22%	0%	22%	19%

Table 3. Adoption Profiles for Demographic Variables

Although small sample size of each group precludes statistical inference of adoption patterns, some preliminary observations can be made from the data in Tables 2 and 3. For example, Table 2 shows that most respondents who had adopted MDA were in CS departments; respondents who were unaware of MDA tended to be in IS departments. Institutions' emphasis on research and teaching and degree programs offered seemed to be somewhat equivalent across most adoption stages. Table 3 shows that tenured and untenured faculty respondents were spread across adoption stages, while non-tenure track and graduate student respondents were more concentrated at earlier stages of the adoption cycle. In terms of courses taught, respondents also tend to span adoption stages, with roughly equivalent percentages at either end of the spectrum. Approximately one third of respondents who teach systems analysis and design (with or without programming) were classified as adopters of MDA, with most of the balance at least at the interest stage. Thus, it does seem that MDA is starting to emerge as an important part of these courses, although the adoption trend clearly has not yet reached a mature stage.

In a formal bivariate correlation analysis using Spearman's ρ, the stage of MDA correlated with adopter characteristics. As anticipated, the stage of MDA adoption was highly correlated with knowledge of MDA ($\rho = 0.742$; $p<0.001$) and interest in MDA ($\rho = 0.669$; $p<0.001$). However, the importance of teaching (teaching emphasis) was negatively correlated with the stage of adoption ($\rho = -0.280$; $p=0.035$); those who rated teaching as less important at their institutions tended to be further along in the adoption lifecycle. Stage of adoption did not significantly correlate with the importance of research at the institution.

USE OF MDA IN TEACHING

When respondents were asked why they teach MDA, they gave reasons that fell into one of four categories: industry trend to adopt MDA, important to teach modeling, improved productivity of students and instructors, and uncategorized responses. Of the twenty responses (one subject made two, unrelated responses), the most (eight) were from instructors who felt that modeling was an important topic in its own right. The next most common response (five responses) was the importance to industry. In addition to the different reasons that instructors teach MDA, their students have had different reactions. While the reactions of their students have been very positive overall, many noted two difficulties that students had in learning MDA. First, many students struggle to learn how to model systems using any paradigm and so MDA's modeling emphasis made it difficult for them. Second, instructors noted that few students appreciate the power of modeling in system building — they often see modeling as peripheral to the job of producing code. In addition, students seldom see the kind of large-scale development projects where the overhead of modeling is paid back with reduced implementation and maintenance effort.

Teaching and learning modeling is a difficult task because it requires judgment and experience to do it well. While there is much to know to write code well, students and instructors are often most comfortable concentrating on the syntax and semantics of a programming language. For most students, being asked to memorize syntactically valid expressions falls in line with their prior school experiences and they receive immediate feedback from the compiler when they're wrong. For instructors, the standard lecture format works well for presenting the material: "This is an *if* statement and it has the following syntax; note that Java does not use an explicit *then* keyword as some other languages do." Learning how to model effectively is much more challenging for students and teachers. One of the long-running challenges of teaching system analysis and design is addressing just how to do it; the existence of model-driven development simply drives the modeling task ever deeper into the software development process and highlights the difficulty even more.

The observation that students fail to appreciate the power of modeling is related to what we call *code worship*, where students see source code as the highest technical expression of information technology. While not all students want to *write* code, even non-code writers see code as the ultimate artifact and assess each other's technical ability based on coding skill — shifting the focus to modeling runs against this drive to be closer to the code. The very existence of MDA, however, argues for the importance of modeling in the development process; indeed, MDA makes modeling the *focus* of the development process. The role of the programmer is to then craft the code that tools cannot automatically generate and much of the system builds directly on models. While students sometimes doubt the opinions voiced by their instructors, new tools and industry trends that they can read about are much more persuasive. Even for those instructors who are philosophically opposed to MDA (who often argue that it's simply the latest incarnation of the dreaded big up-front design, known as *BUFD* in agile circles) MDA offers a chance to discuss the role of modeling in the broader context of software development (Fenstermacher, 2004). More importantly, MDA offers a role for modeling throughout the software development process, which is a welcome alternative to the "model first, code next, forget the model last" approaches that led to the current agile backlash.

LIMITATIONS AND CONCLUSION

The findings from this study indicate that although MDA is gaining recognition and use in academic circles, its diffusion is still somewhat limited. This study is a first step in understanding how MDA is being diffused as a teaching tool in academic institutions. Some shortcomings of this research include the relatively small sample size and the fact that some questions that seek to understand the stage of adoption may not be an accurate reflection of the intent to use MDA in an educational setting. While we may not have a perfect understanding of the diffusion of MDA in academia, this survey is the first of its kind in assessing the current state. It is hoped that this research stream will yield important insights into how technologies such as MDA are diffused in academic environments.

REFERENCES

1. Brassard, M., and Cardinal, M. Addressing Problems with Model Driven Architecture (2002).
2. Davis, G.B., Gorgone, J.T., Couger, J.D., Feinstein, D.L., and Longenecker, J., H.E. (Eds) (1997) IS'97 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, *The Data Base for Advances in Information Systems* 28, 1.
3. Fenstermacher, K.D. (2004) If I had a model, I'd model it in the mornin', *Companion To the 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '04)*, ACM Press, Vancouver, BC, CANADA, 88-95.
4. Fichman, R.G. (2000) The diffusion and assimilation of information technology innovations, in R.W. Zmud (ed.), *Framing the Domains of IT Management: Projecting the Future Through the Past*, Pinnaflex Educational Resources, Inc., Cincinnati, OH, 105-128.
5. Gallivan, M., Truex, D.P., and Kvansky, L. (2002) An Analysis of the Changing Demand Patterns for Information Technology Professionals, *SIGCPR 2002*, Kristiansand, Norway, 1-13.
6. Hagerstrand, T. (1953) Innovations for Luppet Ur Korolgisk Synpunk, University of Lund, Lund, Sweden.
7. Kwon, T.H., and Zmud, R.W. (1987) Unifying the fragmented models of information systems implementation, in R.J. Boland and R.A. Hirschheim (eds.), *Critical Issues in Information Systems Research*, John Wiley & Sons, Chichester, New York, Brisbane, Toronto, Singapore.
8. Larsen, T.J., and McGuire, E. (eds.) (1998) *Information Systems Innovation and Diffusion: Issues and Directions*. Idea Group Publishing, Hershey, Pa.
9. Mawhinney, C.H., Morrell, J.S., Morris, G.J., and Helms, S. (1995) Updating the IS Curriculum: Student perceptions of industry needs, *Special Interest Group on Computer Personnel Research Annual Conference*, ACM, Nashville, TN, 233-234.
10. Mawhinney, C.H., Morrell, J.S., Morris, G.J., and Munro, S.R. (1999) Updating the IS Curriculum: Faculty perceptions of industry needs, *Special Interest Group on Computer Personnel Research Annual Conference*, ACM, New Orleans, LA, 219-221.
11. Meservy, T.O., and Fenstermacher, K.D. (2005) Transforming Software Development: An MDA Road Map, *Computer* 38, 9, 52-58.
12. Misic, M.M., and Russo, N.L. (1999) An assessment of systems analysis and design courses, *Journal of Systems and Software* 45, 197-202.
13. Noll, C.L., and Wilkins, M. (2002) Critical Skills of IS Professionals: A Model for Curriculum Development, *Journal of Information Technology Education* 1, 3, 143-154.
14. Object Management Group, I. (2001) MDA Guide Version 1.0.1.
15. Rogers, E.M. (1995) Diffusion of Innovations, Free Press, New York.
16. Sirosky, C. (2003) Realizing the Benefits of MDA, *OMG Workshop*.
17. Weiszmann, C., and Messenheimer, S. (2005) What's Holding Back OMG's Model Driven Architecture? *Software Development Times*, 128, 28-30.