Causal Mapping for Research in Information Technology

V.K. Narayanan
Drexel University, USA

Deborah J. Armstrong
University of Arkansas, USA

IDEA GROUP PUBLISHING
Hershey • London • Melbourne • Singapore
2005

Chapter X

Causal Mapping for the Investigation of the Adoption of UML in Information Technology Project Development[1]

Tor J. Larsen
Norwegian School of Management, Norway

Fred Niederman
Saint Louis University, USA

Abstract

This research project gathered data about the use of UML and object-oriented analysis and design as the approach to the development of information systems. The data collection method consisted of interviews with information systems application developers with wide ranging differences in background. The authors used causal mapping for analysis of the data gathered. This chapter focuses on the authors' experiences with causal mapping as a method for exploring issues and relationships. Causal mapping was also used to document tips on its use illustrating these with findings regarding UML and object-oriented analysis and design in particular.

Introduction

Productivity in computing hardware for decades has roughly followed Moore's Law in producing doubled power at lower cost every 1.5 years or so. Similarly, the productivity associated with networking technology continues growing exponentially as each new user added creates value for all previously installed users. Only in the arena of software development does productivity seem to be progressing slowly — if at all.

One target area for improving software development productivity has been documentation and requirements structuring. For example data flow diagrams (De Marco, 1978), data modeling (Chen, 1976), and the object-oriented modeling (Brown, 2002) have all been added to the repertoire of systems designers. The underlying concept is that visual representation, accuracy, and a fairly straightforward nomenclature in modeling system characteristics can serve to help bridge understandings among system users, developers, and programmers. Such understandings should allow for reducing the number of systems that are technically valid but don't address business problems and should provide clarity for technical designers and coders to more efficiently translate requirements into artifacts.

Despite the contributions to increased software quality because of the employment of these techniques for representation, problems remain (Sauer, 1999). It is reported that only 12 percent of information system projects are delivered on time and on budget. The most often mentioned reasons for failure are not meeting users' requirements, impaired functionality, and purely technical problems.

Based on these observations, the objective of the present research was formulated as increasing our understanding of how the phenomenon of representation is managed and used in today's business organizations; when does it work; when does it not; what are reasons for its successful or failed employment? We realized that these issues are not only related to projects but also to corporate efforts to define and standardize approaches to software development, education, and outsourcing — just to mention a few dimensions. Techniques for documentation and requirements structuring must be understood in an organizational context.

In this study, we focused on one particular approach to system representation, the Unified Modeling Language (UML) (Booch, Rumbaugh & Jacobson 1999). UML is the most recent among approaches to representation and it is the most complete approach spanning from user-information processes to implementation concerns. It is also widely held to be the future approach to modeling information systems.

The present chapter tells the story of our initial efforts to understand the use of UML in an organizational context. We don't present a traditional report on completed research. Rather, we describe and discuss the issues addressed and the decisions made in our early search for theory, why a data driven approach may be appropriate, why causal mapping was chosen as the method of analysis, emerging results, and lessons learned from this first attempt at creating some kind of order in a highly complex and disjointed research area. The chapter proceeds with theory and research approach. Next follows methodology, followed by material about developing causal maps, leading into challenges in using causal maps and lessons learned. The last section is the conclusion.

Theory and Research Approach

Research and practice reports on UML are diverse and scattered (Cho & Kim, 2001, 2002; Sim & Wright, 2001, 2002; Johnson, 2002). We find documentation of negative as well as positive effects of UML deployment. Also, UML — while studied and described in many computer science papers — has generally not been studied in the evaluative sense: what is its economic contribution to a firm? How does the precision of description it provides trade off against the time it takes to develop and maintain? What is its cost through the lifetime of an application and for use throughout an organizational department? Most profoundly, in our view the established literature typically addresses only a small subset of UML-related issues within a very narrow area of the project effort. Reflecting upon how UML is used in the organizational context, it quickly became apparent that decisions and implementation at multiple levels must be involved. Clearly the organization must permit, if not encourage, its use. Particular projects may subscribe or not subscribe to the UML approach — perhaps for engineering reasons such as a project whose mission is not deemed consistent with this approach or for human resource reasons such as it would take too long to acquire UML development skills to allow the project to be completed on time. We came firmly to believe that use of UML requires the resolution of organizational, project, group, and individual issues. With the exception of textbooks, contributions addressing UML in this larger context could not be found. The challenge of conducting research on UML in an organizational context is further discussed in the subsections of search for a theory, theory-driven versus data-driven research, and use of causal mapping.

Search for a Theory

The lack of an integrative perspective in UML research may not be typical in research addressing similar research objectives. A review of published research within the topic area of management information systems identified diffusion theory as a likely candidate (Fichman, 2000). If we consider UML as a "technology" in the broad sense of a technique and approach to achieving a set of results, the way that it is diffused and adopted among organizations, projects, groups, and individuals would perhaps be a reasonable framework for attempting to address UML usage and the impact it has on project outcomes.

Rogers' (1995) formulation, sometimes called Rogers' theory, has been fairly thoroughly examined using highly precise and rigorous measures — generally gathered through widely distributed survey questionnaires (Brancheau & Wetherbe, 1990; Moore & Benbasat, 1991; Prescott & Conger, 1995; Fichman & Kemerer, 1999). Rogers' theory has also been extensively criticized (Larsen, 2001; Lyytinen & Damsgaard, 2001). For example, organizational IS/IT innovations are more complex than Rogers' theory specifies. The IS/IT innovation processes unfolding in the "adopting" organization are richer and more diverse than sigmoidal. The division of users into the categories of early adopters, early majority, late majority, and laggards is at best unproven but more likely an introduction of social complexity, yet simplicity, that implies semantic meaning contrary to innovation process realities (Van de Ven, Polley, Garud & Venkataraman, 1999). Decisions are not purely individual but distributed among individual actors, groups, project managers, and organizational units. We concluded that using diffusion theory as the platform for studying UML in an organizational context would be inappropriate.

Our concern parallels the debate that has appeared in venues ranging from top journals to informal discussion groups about the "rigor" versus "relevance" and "theory" versus "description" characteristics of research studies (for example, Robey & Markus, 1998; Benbasat & Zmud, 1999; Nurminen & Eriksson, 1999; Goles & Hirscheim, 2001; Weber, 2003). Given how little knowledge links UML to organizational outcomes, the study of UML would suggest "relevance" and "description." At least among the organizations we ended up visiting, the use of UML remains either a new technology that could potentially be adopted or one that is partially in use with the jury still out regarding its precise cost-benefit results. Therefore, adding to the knowledge of UML in the organization would potentially be of benefit to these practitioners. In order to integrate our observations regarding the research characteristics of UML with theories of diffusion, we decided to take a theory building approach.

Theory-Driven vs. Data-Driven Research

We found ourselves in a predicament — staying theoretically pure or reporting on the use of object-oriented analysis and design as it unfolds in organizations today. We came to believe that there is a dialectic struggle between theory and practice (Van de Ven & Poole, 1995). The employment of theory requires a precise definition of constructs, variables, relationships, and units of analysis, or has these elements as the planned outcome of the research effort. Conversely, increasing our understanding of a complex phenomenon, such as the use of object-oriented design and analysis, requires a multilevel research design where the outcome cannot be predicted based on available research reports (Klein & Kozlowski, 2000).

Our research focus on increasing our understanding of UML in its broad organizational setting suggests that many loose ends and disconnected bridges would emerge. The indication is that chaos rules. Although this might be expected, we also believe that some degree of order will exist. In organizational settings, actors allocate time and resources to issues and phenomena that they believe have importance. A high demand for resources in defining and executing action related to the issues and phenomena addressed, as is the case with UML, will increase the likelihood that actors take it seriously. It is reasonable to suggest that actors will concentrate on particular issues and phenomena that they deem as being important. Directing attention to issues and phenomena is a necessary but not sufficient prerequisite for understanding why actors would pay attention. We also believe that actors are concerned about outcome. That is, when time is spent on defining issues and phenomena, attention would also be given to the impact that one of these may have on other issues and phenomena.

Causal Mapping

Our search for an analysis approach that would assist this reality led us to causal mapping. The purpose in selecting causal mapping is not oriented toward defining central tendencies in the data, but rather toward documenting the range of possibilities. In other words, we believe it is too early to test the relative frequency of three theorized causes of UML success when we expect there may be four, five, six, or more potential causes that have not yet been documented. We would consider it too early to even attempt to conclude that in some precise percentage of situations UML increased positive development outcomes by some specific amount. Rather, we use causal mapping to illustrate the range of variables that could influence decisions to implement UML as well as the outcomes should UML be implemented.

The employment of causal mapping in our research setting would differ from most others. For example, Nelson et al. (2000) use the method to define a fairly narrow set of constructs, variables, relationships, and relationship strengths. Although we expected to find constructs, variables and relationships, it is not our goal to demonstrate that our causal map analysis outcome would be reliable across research settings. Rather, we were looking for indications of these as they emerge from the data. Also, we believe that causal mapping would document missing connections (e.g., sometimes it is valuable to know what elements should be absent as well as present in an equation) and bring to light relationships that may not be linear, where a construct at one level may cause negative results, at another level have no impact, and at still another level cause positive results. In conclusion, although we expected to find the same elements as in other research employing causal mapping our objective was not reliability but validity with regard to mirroring the complexity of organizational approach to UML.

Our use of causal mapping has similarity with Fahey and Narayanan (1989). We seek to explore a complex phenomenon (UML) in the wider organizational setting. Our use of causal mapping can be understood in the context of the basic research method challenge of the personal pronouns of we, I, and you (Bohman, 2000). "We" represents the objective approach to research — as in a deductive approach. "I" represents the individualized approach to research — as in an inductive approach. "You" represents the real world subjects and their views, attitudes or behaviors. Obviously, researchers using the voice of "we" or "I" attempt to express salient aspects of "you." Additionally, researchers employ a method within which aspects of "you" are studied and analyzed.

The particular concern in this research was that causal mapping strongly emphasizes a clear definition of the two elements of construct/variable and causal relationship between constructs. Used in a straightforward and narrow manner causal mapping represents the voice of "we" — a deductive and objective approach to research, as in Nelson et al. (2000). The present researchers argue that subjects — the "you" in research — may possess clear views of constructs/variables and causal relationships. However, the presumption, in Nelson et al.'s (2000) research is that these will be parsimonious on the individual level and across individual roles. We believe, however, that our domain of interest, UML within the organizational context, does not have the pre-existing background of prior investigation to allow for identification of those constructs/

variables and causal relationships before examining data, but rather using the data as the source for discovery of these.

Therefore, our data collection method was not intended to define a set of constructs/variables and causal relationships, as in Fahey & Narayanan (1989). Doing so would have the danger in leading respondents to fill in the suggested topics without considering whether the set of topics is complete. That is, since very many constructs/variables and relationships exist in the real world, directing subjects to talk about a pre-selected subset of these phenomena may lead to definitions that reflect the pre-selection rather than represent the real world. In this scenario, the outcome might appear clean, but actually hide the fact of missing elements. Therefore, given the lack of prior research guidance, in this research subjects were encouraged to talk freely while guided toward the present research focus. Our approach should and must be closer to the spirit of having "you" talk as "you" find appropriate. Data collected in this manner would, hopefully, be a good source for defining constructs/variables and causal relationships with a high degree of validity.

Methodology

The first subsection addresses sample definition issues. The thinking behind data collection and lessons learned from it are presented in the subsections of interview guide development, interview execution, and lessons learned from conducting the interviews. The last subsection addresses issues related to coding procedures.

Sample Definition

Organizations likely to meet the qualifications below were asked to participate by inviting volunteers to participate in structured interviews. To qualify as a research participant, the systems development project participants must have had at least three years of work experience and they must have worked on information systems development projects of at least a minimum threshold of size. It was our judgment that very small projects would not require formal documentation. Without knowing an exact threshold where such documentation is needed (in fact one sub-goal of the study is to consider this issue) we projected a minimum of having participated in at least one project involving three full-time project members lasting at least six months. Otherwise the probability of need for formal methods such as UML was viewed as being below an acceptable threshold.

The strategy for selecting participants involved seeking a broad range of systems analysts working on IT development projects in a wide variety of roles. The selection of individuals playing a wide range of roles is based on the primary goal of developing a range of possible states or values for various aspects of UML's role in the development process. Participation was solicited in two ways. First, individuals of personal acquaintance to the researchers meeting the qualifications discussed next were invited to participant. Second, we requested referrals to eligible individuals from organizations likely to have these individuals in their employ. Direct contact with eligible individuals as then initiated.

Although we had a clear description of desired interviewees, we found the need to adjust as opportunities presented themselves. We sought systems developers but ended up with individuals playing a wide array of roles pertaining to system development including corporate strategic information systems planners, project managers, user security officers, information systems consultants, and about 40% systems developers (see Appendix A for a full exposition of the characteristics of the interviewees). This resulted from our collaborators in participating organizations' interpreting our research agenda and the qualifications of likely subjects differently from us. In retrospect, this turned out to enrich the range of perspectives represented in our sample.

Interview Guide Development

As we decided to collect our data using interviews, it was important to develop an interview guide in order to structure the information gathering process. Given that our goal was the solicitation of maximum ideas and insights, we did not expect that each participant follow the identical sequence of questions. However, we did not want to inadvertently miss areas of inquiry with any of the participants.

Questions in the interview guide were mainly derived from our own experience. Our first step involved the development of a "rich picture" (Checkland & Scholes, 1990). It included major relationships among organizational units that may determine strategic issues related to representation, formal educational units, systems development projects (their managers, systems analysts, and programmers), external to the organization units (consultants and vendors), and representation documentation. Literature was also used, for example Booch, Jacobson and Rumbaug (1999) for UML-related issues, Rogers (1995) on diffusion issues, Van de Ven, Polley, Garud and Venkataramann (1999) on innovation issues, but also published research articles on UML. Questions were not copied directly from these or any other source. Rather, these sources were used as a means for checking type and breadth of questions to be included in the interview guide.

In the end, the interview protocol went through nine iterations in its development as we reviewed, combined, separated, and added new questions while tweaking wording of those we intended to use. As part of the development process we read the questions aloud to each other and simulated the sort of responses we expected — looking for alternative interpretations and elements that needed further clarification or terms that needed further definition. We also considered whether the simulated answers would lead to the kind of information that we would want to analyze in later stages in the research project.

In the final version of the interview protocol we had seven sections: background and initiation (Appendix B); description of the most recent IT development project they had worked on; methods and tools used in the most recent project; outcomes they observed from the most recent project; observations about projects within the organization in general; the IT environment in their firm; and the firm's IT strategy. The logic for this sequencing of questions was based on movement from the smallest to largest unit of analysis. We believed that the respondents would be most familiar with their personal experience and concrete observations. As the interview continued, more generalization and global assessments would follow.

Interview Execution

The main approach taken was to conduct interviews at the interviewee's site. The main arguments for this approach was that by doing so interviewees would feel more comfortable with being interviewed and would not have to waste time on travel in a busy work schedule. Nevertheless, four of the interviewees preferred to have the interviews conducted on campus and another at a local coffee shop.

Due to the length of the overall interview, only the first five sections in the interview guide (see Appendix B) were fully covered in all interviews. Within the topical segments, we found the most effective approach was to begin by posing a broad general question intended to gain the participant's uninfluenced view of an area. We then followed up with more specific questions. For example, we would ask a broad general question such as, "What models or approaches did you use for documenting our last IT development project?" We would then continue by asking if the participant used use cases models, then asking if they used sequence diagrams, following the list of UML models. We also asked if they used data flow diagrams, data modeling diagrams, and any other models. This procedure allowed us to receive an uninfluenced view from the interviewee, suggesting ideas most salient from their point of view, while still investigating each of the specific tools comprising UML and, thereby, gaining more detailed insight into the adoption and "customization" of their use of UML toward providing a richer picture of the nature and not just quantity of usage.

Lessons Learned from Conducting the Interviews

We found the interview protocol to be invaluable in two respects. First, during the sessions, it provided a stable background to move through various items of interest with respondents. On occasions when the discussion would go into detail in an area, it gave us a point of reference to return to from which we could continue in another area. Second, in examining our data after completing the interviews and transcriptions, it provided a framework of issues that were addressed in one way or another by each of the interviewees allowing for more systematic comparison of answers in light of differing roles within projects and other differentiating demographic elements of the participants.

One decision that presented difficulties pertained to referencing a particular project. We asked participants to think of a particular project in order to make the questions concretely related to actual experience, rather than derived from how they think "it should be". However, in spite of this, we found that respondents often strayed into more general discussions regarding how they used various approaches across a range of projects. Methodological purity might have suggested forcing the participants back to discussion of the one identified project, however, some of the more interesting points were raised in consideration of the comparison among projects. Our observation is that even as the conversation drifted away from a particular project, it remained based on their experiences, though more generalized across a set of projects. We also found that they might pull details from a variety of project experiences — for example, they might report on their experience with sequence diagrams on one project where they did not use class diagrams, but report on another project using class diagrams but not sequence diagrams. In retrospect, beginning with a single project was probably a good approach because it tended to keep participants from responding only with generalities. But it was also good to allow the respondents to drift away from that particular project in order to report on the broader array of their experiences. In conclusion, we found that the steps taken to strengthen the voice of "you" (Bohman, 2000) was the correct interview approach.

Table 1. Example of the analysis table

C-#	R	P-#	Statement	Cause	Effect
31	2	4	We're still trying to get people to understand the IS model and our roles because people look at us when we say we need to be part of their planning units and their teams and so for, they look at us "We're not ready for IT yet." We say, "No, no, you don't understand. We're a part of your team. We need to understand prior to when you think you need technology so that we can best help you. It also helps us in our planning." So we've got varying degrees of people understanding our role but it is starting to grasp on. People are starting to understand that role and I can't seem to have enough people to fulfill those roles.	Early involvement	User understanding
32	2	4	Also, recently, in the business integration team, we had taken on process mapping, which makes sense because if you don't understand the process, it's difficult to start mapping out a project and mapping out technology. So we are also facilitating and training process mapping so that's also new.	Process mapping	Project organization
33	2	6	We had a team, a fairly large team, that pretty much encompassed a good portion of our IS department internally as well as we brought in external consultants because we didn't feel we had the expertise and knowledge in doing everything that we were going to be embarking upon.	Addition of consultants	Enhance knowledge base for project work
34	2	7	We had some knowledge but it wasn't practical. It was textbook knowledge and we wanted some practical experience so we hired, actually, multiple consultants. We didn't have one firm. We brought in multiple people. We project managed it but we had a project manager from the consulting firm work with us hand-in-hand.	Addition of consultants	Enhance knowledge base for project work
35	2	7	I thought I could because my background seems like I could do that but I very quickly learned that if I got caught too much in the technical details, I couldn't manage the overall scope of the project.	Focus on overview	Project management success
36	2	9	And we went out and personally trained all 100 distributors when we rolled out phase one. They liked that. They thought that was great.	Personalization of training	User satisfaction

Note: The abbreviations in column headings denote: C-# (comment number), R (respondent number), P-# (page number in interview transcript).

Coding Procedures

In general, from this stage on, work was performed by one author and verified by the second author. (It is noted that the reverse was the case for other analyses of the same data). Rather than coding separately then comparing outcomes quantitatively as one does with most communication research coding, we discussed differences to consensus. This method resulted in many observations of differences based more on linguistics than on substantially different understandings. However, we chose this approach because we did not want to start with predefined coding schemes, but rather to generate the language for describing elements based on the content of the interviews themselves.

Coding occurred in two stages. First, we went through each transcript and identified causal statements (or inferred causal statements). These statements were copied and pasted into a table (see Table 1).

The table structure was designed to serve as a tool for organizing further coding efforts. The instantiated tables also provided all relevant information about each statement in a handy and easy-to-use manner. In the end, the transcripts were transformed into a set of tables — one for each transcript.

Second, for each statement we identified "cause" and "effect" labels. Because we examined these at a semantic level — searching for the meaning of the statements rather than keying of specific words or syntactic level structures — we did not develop specific rules, but rather treated each statement as a unique case and applied judgment and discussion to consensus where meanings might have had multiple interpretations. These labels were subject to change in the mapping phase for either of two reasons. First, in the context of the entire interview, the meaning of the particular statement sometimes showed slightly different orientation and second, slight differences in phrasing might be collapsed into a single descriptive term. This occurred often in wording of "effect" variables such as "project success." Slightly different ideas such as project thoroughness, project efficiency, and project learning might all be collapsed into project success. In the end, this process resulted in a total of 270 comments identified from the 11 transcripts (24.4 comments per interview). In addition, some 36 additional cause-effect combinations were observed (where one comment yielded more than one relationship) and another 47 comments contained valuable insights into definition of terms or other matters, but did not yield causal links. In sum, we observed 259 casual links (23.6 links per interview). The complete list of cause concepts and effect concepts are presented in appendices C and D.

Developing Causal Maps

We hasten to say that even though we have accounted for all comments in our current set of causal maps, we continue to find ways of recombining them for the purpose of conducting additional analysis. Following from the verbal description of the content of the statements, visual aids are prepared to help with developing understanding and

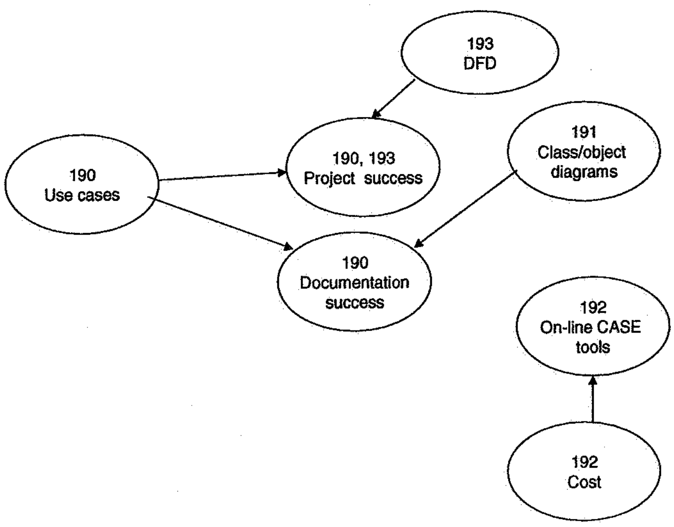
Figure 1. Example of a causal segment

Note: The number in the bubble refers to the statement in the table. These numbers make it much easier to return at a later time to the actual text during the writing stage of the project. Also note that the arrow heads show directionality which is derived from the causal or inferred causal statement. On occasion, arrowheads may go in both directions —for example, developers' skills could influence use of UML, but prolonged use could also change the composition of developers' skills. The arrowheads, therefore, represent the observations of the respondents rather than all relational possibilities.

observation of deeper relationships. This also occurred in several stages. First, we transformed each page of the tables into a set of causal segments — each statement generated a pair (occasionally multiple causal relations) of causal relations which were mapped onto PowerPoint slides. This mapping was done one page of table to one PowerPoint slide. Even at this stage, references to a single entity in multiple states would be represented as multiple arrows to or from that entity. In other words if training influenced user skills and user skills influenced use of UML, the segment would be presented as shown in Figure 1.

Each causal segment represents our smallest unit of result documentation. Each interview would yield a number of causal segments. Some of these might be related and some

Figure 2. Consolidated causal map — Interview #8, documentation success



might not — the latter representing stand-alone findings. The mechanism chosen for combining segments derived from one interview was to look for variables that appear in more than one causal segment. A variable that appears across causal segments can take on the role of being either independent or dependent and sometimes both. Based on these principles causal segments from an interview can be combined into a holistic view of the relationships uncovered in that interview (see Figure 2 for an example).

At this point, we had a choice. We could consider each interview a separate case and treat the discussion of findings in terms of a set of 11 case studies following Yin (1994). However, we opted for the second approach given our focus on theory development and understanding the range rather than central tendencies of the data, which was to further consolidate findings across interview participants. Again, we did this by focusing on "target" labels and searching for the entities that influenced them. This work is in process and will, hopefully, yield insights that are publishable in their own right.

Challenges in Using Causal Mapping

Even if convinced that the method has the potential to yield a quality outcome, ending up with insufficient "revelation" of new or interesting information is possible. Ultimately, one may come away with a study that just doesn't show any consistent findings or even suggest different findings consistently among different contingencies. In principle, this would be an excellent scientific "disconfirmation" of prevailing thought, but practically speaking it is often not a favorite observation in the eyes of reviewers.

However, this problem is often tractable where the researchers have become too close to the project and are either not recognizing what is "interesting" about it or not presenting what is "interesting" in a format highlighting what editors and reviewers will also find interesting. This leads to the question: What constitutes an interesting finding? We reflect on this question by discussing the abandoning of Rogers' diffusion theory, surprising elements, and complex relationships emerging from our causal maps.

Abandoning Rogers' Diffusion Theory

The findings support our view that Rogers diffusion theory is invalid as a lens for studying UML in an organizational context (see Table 2).

Our findings confirm that Rogers' theory does not apply. Our conclusion is restricted to the employment of this theory as the overarching platform for our study. However, our analysis covers major prerequisites within Rogers' theory. Still, other aspects of this theory may be relevant, for example, the constructs of trialability and observability. The proposition is that the higher the degree of trialability and observability, the more likely that adoption occurs. However, UML is in its very nature a highly complex phenomenon — in fact so complex that Rogers definition of these terms quite likely does not have meaning.

Table 2. Rogers' diffusion theory and present findings

Aspect of Rogers' diffusion theory	Findings in the present study
The innovation is clearly defined and does not change during the diffusion process	UML is highly complex in its own right and exists in parallel with other methods for representation. UML consists of multiple elements. UML is differently understood among actors. UML is molded according to need during the systems development process. When standard application packages are employed, UML may be an improbable approach.
Over time, the adoption of the innovation is sigmoidal	Even when a corporate strategic unit promotes the employment of UML, projects in the organization may not use it – rendering a smooth cumulative usage curve quite unlikely. Even the concept of use (of UML) in its most narrow interpretation is a difficult issue. We found that a project manager clearly stated that UML was the standard being used. The systems analysts disconfirmed.
Users can be divided into the categories of early adopters, early majority, late majority, and laggards	Awareness and/or belief in UML occur collectively and individually at differing points in time. Adoption of UML is not necessarily a "good thing." Being a laggard might be highly appropriate.
Individual adopters make the adopt/non-adopt decision	Actors on many levels make decisions about UML. More often than not decisions made are not followed through. The issue of mandatory (because of organizational policy) and voluntary use is highly complex.

We cannot suggest in a straightforward manner what a theory that would describe the phenomenon of UML in an organizational context might be. Therefore, listening to the data, we think, is extremely necessary in our effort to increase our understanding.

Surprising Elements

We look for variables that have not been mentioned in the literature. For example, we find in our causal maps that the implementation of standard application packages is not related to successful use of UML. We found this surprising because many organizations today install an integrated standard application package for critical business functions such as accounting, procurement, inventory control, customer order processing, and distribution. The package must be integrated into the portfolio of other information systems in the organization.

However, the structure and architecture of the standard application package generally is not made available to the buyer organization. The reason for this is that the standard application package vendor looks upon the structure and architecture as a competitive element. The vendor does not want to run the risk of having other vendors learn the inner workings of the standard application package — the risk of which would increase to an unacceptable level if a buyer organization were to learn about these in depth.

Therefore, the buyer organization is left with descriptions of input and output data as the basis for integration with other information systems. It does not matter whether the standard application package is developed fully in UML or not. Because of denied access, the buyer organization cannot (fully) integrate the information systems portfolio according to UML principles. In fact, if installing standard application packages becomes the dominant strategy among organizations, the concept of UML may be rendered obsolete in companies other than those creating the packages. The challenge would move from a consistent development of information systems to the integration among them.

Complex Relationships

This study identified several areas where relationships are complex. For example, our data suggest that the amount of documentation has some causal relationship with the amount of documentation success — however, this is probably not a linear relationship at all values. At some point increasing the documentation by one unit provides less than one unit of additional value but continues to cost one unit of time and money. At some point additional documentation may start to erode documentation success, where success is viewed in cost/benefit terms. It also includes multi-directional relationships. Our data suggests that there is a relationship between the success of individual projects and the success of the management of project development as a portfolio in the organization. However, the direction of this relationship could go either way. The success of the organization's project development might simply be conceived as the accumulation of outcomes from each project — the sum if you will. This might be influenced by overall policies, tools, and approaches, which may differ among three groups of stakeholders, the IT management, IT workforce, and of the larger business influences. On the other hand, these same policies, tools, and approaches at the departmental level may influence the success of each project. Note that policies might be applied with wisdom differentially where appropriate to different projects, or they might be applied uniformly helping some projects and retarding others. Note also that some types of policies, perhaps pertaining to standardization or reuse, would have dramatic impact while others pertaining to documentation style or change management would have less impact given different circumstances and projects.

The nature of these relationships might be difficult to detect from the causal mapping *per se*, but hints can be detected where different interview participants use terminology differently, remark on relationships from different perspectives, point out contingencies, or otherwise describe complexity in their responses to questions. The process of coding statements noting causal elements and effect elements tends to blunt the observation of these "semantic" level observations. However, in the consolidation of maps, these tend to show up as variations among elements described by different individuals. Mining the interesting aspects of these complex relationships requires returning to the original text and also some interpretation, inference, and imagination on the part of the investigator. Imagination isn't a term normally associated with "scientific studies," but in the sense that the investigator recognizes that an interviewee is describing a relationship in a particular context and that the relationships could have different aspects outside of that context by imagining or envisioning alternate scenarios, can help bring out the richness of the phenomena, even if it does extend beyond the literal statements made by the participants.

Lessons Learned

Throughout this chapter, we have tried to note our observations about the process of conducting a causal mapping study. We'd like to highlight lessons we've learned in three major areas in this endeavor; the mechanics of conducting a data driven study; the value of planning; looking for consistency and gaps.

The Mechanics of Conducting a Data-Driven Study

The mechanics of conducting this sort of study are significant — you can't let them become overwhelming. In our case, these mechanics included a major detour of time and effort for satisfying the requirements of the institutional review board (a.k.a., human subject committee). In the end, the effort needed to pass muster with for human subjects research forced us into some early planning efforts that paid off. On the other hand, the overhead of record keeping and drafting warning statements to participants for assuring privacy at times seemed highly disproportional in effort for protecting against the small risk and small probability of any harm coming to a study participant. Other mechanical difficulties included the extraordinary time it took for transcriptions and checking transcription. This included manual coding and mapping (though we don't know how much confidence we would have in any software program that claimed to provide coding or mapping at a semantic or meaningful level), and, in our case, coordinating work by researchers residing each on a different continent.

The Value of Planning

Steps taken early in the process can make life easier (or more difficult) in later phases. In particular, we would note the value of planning for causal mapping while creating the interview protocol. It can be argued that questions should be developed pertaining to the domain of investigation without regard to the particular method of analysis. In essence, this is what we did for this study. However, as a result we probably had to do much more inferring of relationships than if we had designed the questions with causal mapping in mind. Because we are looking for explicit illumination of relationships, it would be very helpful to fill an interview protocol with "why" questions, particularly following from open-ended observations, to elicit these types of answers. It would also be interesting to see where respondents base conclusions on observations or extrapolation themselves without having underlying causal models at play.

Therefore, directly eliciting information about causal relationships, drivers, and outcome may be problematic. People may not necessarily think about their experiences in those terms. Raising the issue may make people overtly cognizant about these issues on the expense of other cognitive patterns — for example, unrelated events, events forming networks with no clear causal relationships, or events perceived as hierarchies. The argument that causal relationships that emerge in talks not having this focus may be more trusted than those expressed because the respondent is queued along this line has merit and warrants explicit consideration.

Looking for Consistency and Gaps

We were looking for consistency in cause and effect within as well as across interview transcripts. We find that the causal mapping clearly documents gaps. We raise two issues. First, people might simply not have a unified view of UML. For example, the lack of an implemented organizational and project UML strategy will quite likely enforce this result. People would be left to make their own decisions. Second, the gaps may indicate lack of or inconsistent data. The number of interviewees in our research is only 11, which may not be sufficient for convergence of findings. Our subjects were spread among projects and organizational units and among organizations. Therefore, gaps in the causal mappings may be used to determine additional need for data collection. What roles should additional subjects have and how many within each role would be needed to increase the validity and reliability of the research findings? Hence, causal mapping may be used as a vehicle for detecting inconsistencies in practice as well as in research.

We believe that our approach to data collection was appropriate. We did not use causal mapping principles as the basis for interview guide design. The objective of data collection was to allow the voice of "you" to speak in a fairly free manner. The interview guide therefore focused on UML issues to allow the respondent to express her or his own views. Causal mapping was employed as a method of analysis of interview transcripts. We conclude that although causal mapping may be looked upon as a tool within the deductive school of research it is also well-suited for qualitative and exploratory purposes.

Conclusion

Our research objective was formulated as understanding representation in systems development in an organizational context. The approach to representation investigated was UML. We did not find a theory that would serve as a lens for defining issues, constructs or relationships. As an example, the often-used diffusion theory (Rogers, 1995) was found to be inappropriate, relative to our research needs. We decided that a data-driven approach, allowing practitioners to a large degree to speak in the voice of "you," should be employed.

Because we think that practitioners will be concerned about UML-related issues and the impact of one issue on another — for example, that successful use of UML in system development would result in projects being completed on time and within budget — we turned to causal mapping as the method for data analysis. Although our research isn't completed, we find that causal mapping served us well in documenting constructs, variables, and relationships that practitioners deem relevant. Moreover, causal maps, as we hoped, resulted in the documentation of surprising elements, as well as gaps.

This chapter has focused on describing the methods utilized in a causal mapping study. The emphasis has been on presenting and discussing the decisions that arose during the process. It is not difficult to explain our thinking regarding the directions we selected at various choice points. It is more difficult to propose that these were "the right" or even good decisions. On the whole many of these decisions represent trade-offs – getting a reasonable job done versus holding out for a theoretical perfection; maintaining a usable audit train versus not getting bogged down in our own detailed documentation; understanding in detail the view of our respondents versus emphasizing more general emergent themes that no individual may have directly expressed.

From our personal point of view, the study has significantly created value in providing us with a much richer understanding of the role of UML in IT development – lessons that are extremely helpful in the classroom and working with recruiters of our students. It is our hope that the ultimate observations that will be presented in a final report will be viewed as combining elements of "confirming" or adding weight to commonly held views that remain basically anecdotal in nature and elements of some surprising and new relationships and possibilities.

Acknowledgments

The authors wish to acknowledge the contribution of Saint Louis University for providing a summer grant to the second author for work on this chapter.

References

Benbasat, I., & Zmud, R.W. (1999). Issues and opinions – Empirical research in information systems: The practice of relevance. *MIS Quarterly*, 23(1), 3-16.

Bohman, J. (2000). The importance of the second person: interpretation, practical knowledge, and normative attitudes. In H.H. Kögler & K.R. Stueber (Eds.), *Empathy & agency: The problem of understanding in the human sciences*. Boulder, CO: Westview.

Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The unified modeling language user guide*. Boston, MA: Addison-Wesley.

Brancheau, J.C., & Wetherbe, J.C. (1990). The adoption of spreadsheet software: Testing innovation diffusion theory in the context of end-user computing. *Information Systems Research*, 1(2), 115-143.

Brown, D.W. (2002). *An introduction to object-oriented analysis: Objects and UML in plain English*. New York: Wiley.

Checkland, P., & Scholes, J. (1990). *Soft systems methodology in action*. Chichester, UK: Wiley.

Chen, P.P.S. (1976). The entity-relationship model – Toward a unified view of data. *ACM Transactions on Database Systems*, 1(1), 9-36.

Cho, I., & Kim, Y.-G. (2001-2002). Critical factors for assimilation of object-oriented programming languages. *Journal of Management Information Systems*, 18(3), 125-156.

De Marco, T. (1978). *Structured analysis and system specification*. New York: Yourdon.

Fahey, L., & Narayanan, V. K. (1989). Linking changes in revealed causal maps and environment: An empirical study. *Journal of Management Studies*, 26(4), 361-378.

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* (pp. 105-127). Cincinnati, OH: Pinnaflex Educational Resources.

Fichman, R.G., & Kemerer, C.F. (1999) The illusory diffusion of innovation: an examination of assimilation gaps. *Information Systems Research*, 10(3), 255-275.

Goles, T., & Hirscheim, R. (2000). The paradigm is dead ... long live the paradigm: the legacy of Burell and Morgan. *Omega*, 28(3), 249-268.

Johnson, R.A. (2002). Object-oriented analysis and design – What does the research say? *Journal of Computer Information Systems*, 42(3), 11-15.

Klein, K., & Kozlowski, S.W. (Eds.) (2000). *Multilevel theory, research, and methods in organizations: Foundations, extensions, and new directions*. San Francisco, CA: Jossey-Bass.

Larsen, T.J. (2001). The phenomenon of diffusion: Red herrings and future promise. In M.A. Ardis & B.L. Marcolin (Eds.), *Proceedings of the IFIP TC8 WG8.6 Fourth Working Conference on Diffusing Software Products and Process Innovation, April 7-10*, (pp. 35-50). Banff, Canada. Boston, MA: Kluwer Academic Publishers.

Lyytinen, K., & Damsgaard, J. (2001). What's wrong with the diffusion of innovation theory? In M.A. Ardis & B.L. Marcolin (Eds.), *Proceedings of the IFIP TC8 WG8.6 Fourth Working Conference on Diffusing Software Products and Process Innovation*, April 7-10, (pp. 173-190), Banff, Canada. Boston, MA: Kluwer Academic Publishers.

Moore, G.C., & Benbasat, I. (1991). Development of an instrument to measure the perceptions of adopting an information technology innovation. *Information Systems Research*, 2(3), 192-222.

Nelson, K. M., Nadkarni, S., Narayanan, V.K., & Ghods, M. (2000). Understanding software operations support expertise: A revealed causal mapping approach. *MIS Quarterly*, 24(3), 475-507.

Nurminen, M.I., & Eriksson, I.V. (1999). Research notes – Information systems research: The 'Infurgic' perspective. *International Journal of Information Management*, 19, 87-94.

Prescott, M.B., & Conger, S.A. (1995). Information technology innovations: A classification by it locus of impact and research approach. *DATA BASE Advances*, 26(2&3), 20-40.

Robey, D., & Markus, M.L. (1998). Beyond rigor and relevance: producing consumable research about information systems. *Information Resources Management Journal*, 11(1), 7-15.

Rogers, E.M. (1995). *Diffusion of innovations*. New York: The Free Press.

Sim, E.R., & Wright, G. (2001-2002). The difficulties of learning object-oriented analysis and design: an exploratory study. *Journal of Computer Information Systems*, 42(2), 95-100.

Sauer, C. (1999). Deciding the future for is failures: not the choice you might think. In W. Currie & B. Galliers (Eds.), *Rethinking management information systems: An interdisciplinary perspective*. Oxford: Oxford University Press.

Van de Ven, A.H., Polley, D.E., Garud, R., & Venkataraman, S. (1999). *The innovation journey*. New York: Oxford University Press.

Van de Ven, A.H., & Poole, M.S. (1995). Explaining development and change in organizations. *Academy of Management Review*, 20(3), 510-540.

Weber, R. (2003). Editor's comment – Still desperately seeking the IT artifact. *MIS Quarterly*, 27(2), iii-xi.

Yin, R.K. (1994). *Case study research: design and methods*. Thousand Oaks, CA: Sage.

Endnote

1 Authors are listed alphabetically and contributed equally to this article

Appendix A: Respondent Demographics

Employer	Background	Age	G	Management Responsibility	Title
Large accounting/consulting firm Reports to senior manager or VP Organized by industry sectors	Bachelors in Engineering 1994, MBA 96 With firm since 1998	29	M	In projects 2 ½ yrs focus on success & risk, analyzing requirements	Project manager
Medium-sized custom manufacturing Reports to IT director new role/division -IT not quite understood reporting track -president is engineer -CFO -director of IS	BS Computer science	40	F	Manages two teams of business consultants Supports sales and e-commerce helpdesk since 09/2001	Business Integration Manager
Medium-sized custom manufacturing Reports to IT director -president is engineer -CFO -director of IS	Bachelors in Information systems	40	F	two years' tenure manages 13 developers	Application systems manager
Large accounting/consulting firm	Bachelors in Philosophy Business certificate MBA -employed since 1999 in current role	31	M	responsibility is in the middle of the hierarchy last project, 8 people reporting directly, four more reporting in-directly	Development manager
Medium-sized financial brokerage IT dept. has 20yrs old IT in a-transition process	Bachelor of Science in electrical engineering	47	F	mentor for OO/JAVA (80%) acts as liaison between IT department and business line IT users	Lead developer

Appendix A (Continued)

Employer	Background	Age	G	Management Responsibility	Title
Medium-sized financial brokerage -not critically dependent upon IT, makes money! -outsource accounting (new) -move toward std packages -IT department -CMM level 0 but not formally measured -Data management Group	Bachelors in theology & philosophy masters of divinity -12 yrs with firm	50	M	soft supervision, no hiring, etc. technical rather than management track, more consulting than directing	senior technical data consultant
Medium-sized health care facility heavy into IT development -Clinical team within IT department	Bachelors of Science in nursing1999 MS nursing, emphasis informatics, 2002	42	F	In charge of project implementation, but no reporting employees	Clinical business analyst
Small biotech firm -IT department	college 1 year 1980, computer science -was employee-rehired as individual consultant	41	M	none	Contract programmer/analyst
Small biotech firm -IT department	BS-comp.science, 1983 Worked in field since that time	41	M	In charge of project and one other employee	Software engineering manager.
Large food product company flattened out budgets -IT department -1200 employees	electrical engineering, 1968	50	M	None Technical consulting	Corporate level software architecture
Large food product company -Management System Group -- Organizational Technical Development	Masters in aerospace engineering	42	M	-15 people recently, down to 10 today (08/2002	-MIS consultant

Appendix B: Interview Guide

Systems Development Representations The Unified Modeling Language: Organizational Prerequisites and Use Value

Preamble

We began with informal explanation of who we are and what we were doing. We also read the formal Institutional Review Board – human subjects – approved disclaimer.

1.0 Background and Initiation

Individual characteristics
(demographics, subjective norms, competencies and preferences, styles)

1.1 Your age? ______________________________
1.2 Your educational background? (degrees earned)
1.3 Do you have management responsibilities? (Are there other employees that report to you/that you supervise?)
1.4 Your place in the organization (IT-department, Line unit)?
1.5 Your present position?
1.6 Your work experience (position, start-end, type of projects)?

2.0 The Last Completed Project Worked on, Being a User Oriented System Lasting More than Three Months and Having at Least Three Project Members

2.1 Please briefly describe the intended objectives and deliverables
2.2 Approximately when was it started and completed?
2.3 How would you describe the size or scope of the project (lines of code, # of screens, # of tables, # of reports, # of servers/clients/networks, etc.)?
2.4 How would you describe the personnel involved in the project (# of IS professionals, # of client team members, amount of change of personnel during the project)?
2.5 How would you describe the level of difficulty of the project (system characteristics, management issues, collaboration tasks)?
2.6 To what extent would you characterize the project as developing completely new application software (in contrast to maintenance and upgrading)? To what extent did you utilize vendor developed software in the project? (If so, which product did you use, who was the vendor, how would you describe the documentation, vendor determined, object-oriented, in compliance with UML, structured design — or any other? Changes in the vendor software were made by yourself, the vendor, both?)

3.0 Methods and Tools in the Last Project

3.1 As you work on the analysis and design of the new IT application, what sort of methods do you use for representing requirements and application structures? As you personally see it, would you say your employed methods belong to types of methods? If so, what would be the dominating type? The least used or important one?

3.2 Have you used any of the following analysis and design modeling techniques for representing requirements and application structures? *(For any used, follow up with how extensively, supported by case tool, used alone or part of project team usage, thoughts about using the tools, where/when in the project were they be employed)*

- Use-case
- Class or object diagrams
- Sequence charts
- Collaboration charts
- Activity diagrams
- State transition diagrams
- Data Flow diagram
- Entity Relationship diagram

3.3 When it comes to modeling the requirements, design, and code structure for new system, what methods do you prefer? *(May be answered in earlier question — particularly if using methods not selected themselves)*

3.4 In the execution of the last project, to what extent did you use computer tools that support your standard — that is, how do you actually carry out your descriptions, manually or automated*? (link to CASE tools)*

3.5 Do all members of the project team use the same set of representations and CASE tools or does each member use whatever tools he or she prefers?

3.6 How are descriptions of requirements communicated among project members? Do you typically use any of the following for communication regarding requirements among project members?

Mail

- E-mail
- Shared project specification database
- Meetings and walkthroughs
- Other?

3.7 How would you characterize the user — user participation — IT expert interaction and integration (or lack thereof)?

Do you employ UML or any of its diagrams in these interactions? (As you see it, what are the benefits and what are the drawbacks? How well do users understand these diagrams? How does a user exert influence on choice of diagram or diagram content? Or should the user not have any influence? Why and why not?)

3.8 Did this project include any mechanism for learning or enhancing your abilities regarding the representations and CASE tools?

In your experience, how are representation and tool competence taken care of? (follow-up) How much do you have to learn on your own? Did you have 'methods' discussion groups? Did you attend methods seminars or courses? Is the focus on representation something resembling a 'one shot' or is it an ongoing activity?

4.0 Outcomes (Use — Amount, Density, Distribution, Economic Value, "Correctness of Solutions")

4.1 Considering these same projects, to what degree do you consider them successful? What sort of criteria do you use to judge them?

- Economic?
- Time?
- Quality?
- Amount of completeness?
- Positive feelings among developers?
- Positive feelings among users/sponsors?
- Meeting stated objectives?

4.2 To what extent and in what way did the use of UML, OO analysis and design, and/or CASE toolaffect the level of success achieved?

5.0 About Project Work in General

UML/OOAD/CASE — the innovation and its nature (Rogers' aspects, critical mass, OOP infusion — supporting tech, class libraries, application components)

5.1 Having discussed the use of methods in the last project — thinking about the projects you have worked on during the last five years, are there some projects for which you tend to use modeling tools rather than others? What would differentiate projects where you do use the tools from those where you don't?

5.2 Thinking back over the last 5 years or so, what are the most drastic changes with regard to how you work on projects? (organization, people, ITs development, methods)

5.3 What, if any, changes have you made to the standard UML/OOAD/CASE approach to make it useful in your environment?

5.4 Professionally, what would you say is most fun to do?

6.0 IT-Environment (org factors, size (6.1), diversity (6.2), scale (6.3)) — (Following Fichman)

6.1 About how many projects will be active at any one time?

How many employees are there in the IT department who are likely to be engaged in software development at any one time?

About what size is the annual IT budget?

6.2 (Diversity) About how many different programming languages used by the development staff account for at least 5% of development projects?

About how many runtime platforms account for at least 5% of new development projects?

6.3 (Learning related scale) What percentage of application effort goes into new development, integration of systems, maintenance, non-development related activities?

7.0 IT Strategy

7.1 To what extent is there an IT strategy linked to organizational business strategy?

7.2 To what extent is there an articulated IT strategy?

Do you have a development and maintenance strategy for tools and technical support?

Does this strategy include organization issues

- In-house vs. outsourcing?
- Reward system?
- Productivity guidelines?
- Project principles?
- Hiring guidelines?

7.3 Has the strategy any formulation about learning? Keeping the present staff up-to-speed, project learning requirements as part of project management, planning, and control?

7.4 To what extent does your strategy include standardization of hardware, software, and development approach?

7.5 To what extent does the department use OO analysis, design, programming?

7.6 Does the strategy include changing the amount of OO?

THANK YOU

Do you want to be apprised of results of the study?

Name ____

Address____

E-mail

____

Appendix C: Concept List for Independent Variables (Cause)

Ability to translate to other language/culture	OO definition
Abstract thinking	OO design
Active document management	OO programming language
Activity diagrams	OO tool use
Adding requirements into package code	OO use
Addition of consultants	Org strategy
Additions to package	Organizational computing outcome measures
Adoption of standards	Outcome measure
Analysis	Outcome measure (defects)
Analysis process, descriptive	Outcome measure (usage)
Application of OO throughout project	Outcome measures
Application type	Package customization
Architect correctly	Paper prototyping
Architecture level	Pattern of modeling content
Asset management	Paying for changes to specs
Attention to user	Performance
Bugs	Personalization of training
CASE tool use	Phased development with multiple leaders
CASE tool use (Rational Rose and Visio)	Physical arrangement
Change management	Procedural programming skills
Change management process	Process mapping
Change processes	Process time
Class diagrams	Project domain
Class models	Project management
Class or Object Diagrams	Project manager people skills
Clauses where requirements not fulfilled	Project manager technical skills
Communication	Project objective, measure of success
Communication	Project outcome measures
Conflicting assignments	Project size
Consistent modeling	Project size
Consulting firm	Project staffing
Cost	Project staffing
Cost center	Project success measure
Cost per transaction	Prototypes
Culture	Prototyping
Customer knowledge base	Quality assurance
Data modeling	Quality of architecture
Data services layer	Quality of contact with users
Database tool	Rapid evolution, many changes
Decentralization	Rational Rose use
Defined requirements	Rational Rose/UML
Demands on staff versus actual capabilities	Reduction of technical "holds"
Description of analysis process	Relationship of DFD to ER modeling
Description of anticipated development environment	Requirement specification
Descriptive, communication tool	Requirements
Design processes	Requirements determination staffing
Developer acceptance	Requirements Gathering
Developer coordination	Reusability
Developer preference	Reuse
Developer skills	Roles of implementation and domain models
Development environment	Scalability
Development staff skills	Scope creep
Development tasks	Self-training

Appendix C (Continued)

Documentation consistency	Specifying outcomes
Documentation detail	Staff communication between project phases
Documentation method	Staff knowledge and skill
Documentation process description	Staff preferences
Documentation quality	Staff skills
Documentation utility	Staff turnover
Early involvement	Staffing
Early testing	Staffing turnover
Enterprise-level requirements process details	Standardization
ER (data) model	Standardized platform approach
ER diagrams	Standardized use
Faith and trust	Standards
Focus on overview	Success measures
Formal information requirements	Task accounting
Functional area	Task complexity
Guideline design/process	Team size
Hardware capacity	Technology diversity
Hardware/software capacity	Testing/quality assurance
Hierarchy chart	Time, money
Hiring	Tool investment
Ideal documentation approach	Tool platform selection
Implementation issues with ER versus OO	Tool use
Implementation of modeling	Tools
Individual expertise and contribution	Tools description
Individual performance	Training
Information requirements documents	Training methods
IT staff skill levels	UML
Java provides some CASE tool functionality	UML use
Java, OO approach	UML/CASE tools
Leadership	Uncertain situations
Learning / Improvement	Universal development approach
Level at which business process documentation occurs	Use
Linkage of requirements to technical models	Use case
Management mandate	Use of OO
Manual approach	Use of packages
Master scheduling	Use of UML
Matrix organization	Use/non-use of UML
Measurement	User characteristics
Meeting all requirements	User contract
Mentors	User expectations
Metrics	User involvement
Modeling	User liaison
Modeling formality	User satisfaction
Modeling thoroughness	User signoff
Modular training	User survey
Multinational staffing	User type
Narrowed features/simplification	Value of CASE tools, difficult to measure
Need for developer – user communication	Vendor experience
Non-OO design	Vendor selection
Object /class diagrams	Vendor staff turnover
Object diagrams	Visual representation
Object-ERD	Visualization of screens and outputs
On-line CASE tools	Word document
OO	Work expectations
OO Approach	--------------------

Appendix D: Concept List for Dependent Variables (Effect)

Ability to model system	Project cost
Ability to provide documentation	Project difficulty
Ability to provide modeling	Project management organization
Adoption of UML practices	Project management quality
Amount of risk	Project management success
Application success	Project organization
Change management	Project outcomes
Client satisfaction	Project process success (ease/flow)
Client satisfaction	Project quality
Communication requirements	Project success
Communication with users	Prototyping
Component oriented production environment	Quality assurance
Cost	Quality of documentation
Design quality	Quality of information requirements document/
Detailed user requirements	Quality of packaged processes
Developer satisfaction	Quality of tool use
Developer skills	Quality of use of OO
Developer team communication	Quick view
Developer-user communication	Requirements documents
Development guidelines	Reuse
Development model	Role of analysts
Development outcome	Role specialization
Deviation from standards	Satisfaction measure -- questionnaire
Documentation	Scalability
Documentation outcomes	Scope definition
Documentation quality	Scope problems
Documentation success	Sense of closure
Documentation utility	Skill development
Documenting business issues/decisions	Skills
Ease of data retrieval for user	Staff mentoring
Ease of learning OO	Staff skills
Economic value	Staff skills and knowledge
Effective tool use	Staff training
Effort on documentation	Staffing alternatives
Enhance knowledge base for project work	Standardization
ER use	Standardized platform approach
Extra overhead	Standardized use
Formality of documentation	State chart diagram
Formality of modeling	State transition diagrams
Generating user feedback	Subset of CASE tools
Information requirements success	Success measure – fulfill all contracted requirements
In-house use of CASE tool	Sufficient modeling
Insurance against personnel loss	System consistency
Integrating data and process views	System use
Integration of development environments	Technology diversity
Internal versus external staffing	Technology diversity
IT role	Tendency to use OO/UML
Learning curve	Testing effectiveness
Maintenance	Thorough modeling
Model use	Tool use
Modeling formality	Training
Narrowed features/Simplification	Turnover
Need for analysis and documentation	UML use

Appendix D (Continued)

Need for requirements documentation	UML use (sequence diagrams and use case diagrams)
Need for specific tool	Use case
Obligation satisfied	Use levels of ER
On-line CASE tools	Use of DFD and ER
OO	Use of modeling
OO project management	Use of OO approach
OO skill development	Use of sequence diagrams
OO skills	Use of use case
OO success	Used for development
OO tool use	Used for testing
OO use	Usefulness of models
Package customization	User participation
Package stability	User requirements writing
Pattern of modeling content	User satisfaction
Performance	User understanding
Pressure for IT personnel to perform	Version control
Problem understanding	Visualization of documentation