Chapter X

Causal Mapping for the Investigation of the Adoption of UML in Information Technology Project Development1

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.

This chapter appears in the book Causal Mapping for Research in Information Technology by V. K. Narayanan and Deborah J. Armstrong. Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
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.
A Particle Swarm Optimization Approach to Fuzzy Case-based Reasoning in the Framework of Collaborative Filtering
www.igi-global.com/article/a-particle-swarm-optimization-approach-to-fuzzy-case-based-reasoning-in-the-framework-of-collaborative-filtering/111312?camid=4v1a