Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1998 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1998

Maintenance Activities in OOP and 3GL

Michael Eierman
University of Wisconsin Oshkosh

Mark Dishaw
University of Wisconsin Oshkosh

Maintenance Activities in OOP and 3GL

Michael A. Eierman
Mark T. Dishaw
College of Business Administration
University of Wisconsin Oshkosh

Introduction

Maintenance is the process of changing existing software to modify software, either to add or enhance existing functions, or to fix errors (bugs). Maintenance is the dominant activity of many MIS organizations. An average of 70% of software budgets are devoted to maintenance (Swanson & Beath, 1989).

Object-oriented programming (OOP) languages have been in used in research and, to a limited extent, in practice for approximately fifteen years. However, it was not until recently that OOP received considerable attention from the practitioner community. The object-oriented paradigm has been hailed because of its natural relationship to the real world, ease and speed of development, and ability to reuse objects in different programs. These capabilities are thought to produce major productivity gains over programming in 3GLs. However, given the importance of maintenance in the MIS organization, the decision to develop systems using OOP must include evaluation of the relative effectiveness of maintenance in that environment. Unfortunately, little research investigates the maintainability of applications written in object-oriented languages. The objective of this study is to examine the maintenance of systems developed in objected-oriented programming languages.

Research Model

The research model for the maintenance process is developed based on Vessey's Software Maintenance Activity Model (Vessey,1986), and the program understanding literature. Basic software maintenance tasks in the model include Understanding, and Transformation.

Vessey (1986) divides maintenance debugging activities into three categories. These are planning activities, knowledge building activities, and "bug-related" activities. Planning activities center around the management of the task and the coordination of other activities towards the completion of the task. Knowledge building activities center around the procedures involving the acquisition and management of information needed to accomplish the goal at hand. Knowledge retrieval is the accessing of technical information regarding the language of the program or related system software from memory or from other external sources. "Bug-Related" activities in the Vessey maintenance activity model center around three types of actions, clue generation and management, hypothesis generation and evaluation, and error management.

The understanding task, as described above is composed of three sub-tasks or activities: planning, knowledge building, and bug-related (diagnostic) activities. The key, as noted above, to understanding is the development of an appropriate representation and its manipulation.

Program understanding is not the entire story of software maintenance support. The programmer must be able to actually change software and document the effects of that change. The change cycle includes making a change in a source module, compiling the module, and testing of the changed program. These activities are considered Program Transformation.

Research Method

This research examines several propositions about the nature of maintenance tasks in object-oriented programming versus the nature of maintenance in 3GL's. Smalltalk represents the object-oriented paradigm in this research while Cobol represents the 3GL paradigm. The basic unit of analysis for this study is the individual maintenance project. Project, in the context of this study, is taken to mean a change to an existing function, or group of related functions that can be accomplished by a single programmer analyst.

The first study in this research project employed the field study method. The subjects for this study were working programmer analysts performing maintenance tasks. The projects included in the study were selected from the existing maintenance backlog. The data for this study was collected via questionnaire from organizations doing maintenance in the appropriate language.

The second study will collect the same data with the same method. However, the subjects of the study are students doing a maintenance project in Smalltalk and COBOL classes at our university. The final assignment in each class is a project to modify a program presented by the instructors. The program in each class will be written in the appropriate language but provide the same function. This second study is necessary due to the inability to collect field data for Smalltalk from more than one organization. The results of the first study are not conclusive because we cannot rule out an organizational confounding in the results.

Research Questions

Is there a difference in the activities required to maintain applications written in object-oriented languages and those written in third-generation languages? This research question and the model of the maintenance task is used to generate several propositions. The research model consists of four types of activities divided in two classes. The first class is Understanding, consisting of Bug-Related Activities, Knowledge Building, and Planning activities. The second class is Transformation. The primary proposition concerns the difference in maintenance activities between OO and 3GL environments.

Proposition 1: The proportion of maintenance sub-activities is the same for maintenance in OO and 3GL environments.

In either environment, the programmer must understand, plan, and coordinate their changes to the program with other programmers before they make actual changes in the code. However, as noted in the discussion of the maintenance process, programs with "de-localized plans" are more difficult to understand for programmers (Letovsky & Soloway, 1986). Also, complex systems with many modules and links to other systems invite errors of omission and incomplete understanding (Letovsky & Soloway, 1986).

The object-oriented environment embodies de-localization and interaction with other systems because programming is the linking of independent objects that perform different functions. This suggests that programmers in an OO environment may need to spend more time understanding and coordinating their changes before actual enhancements or bug-fixes can be undertaken. On the other hand, the OO environment is described as being easy to understand because of its natural relationship to the real-world. This may result in less frequent need to undertake planning and knowledge building activities.

Proposition 1a: The proportion of Understanding activities is more for maintenance in the OO environment than the 3GL environment.

Proposition 1b: The proportion of Transformation activities is less for maintenance in the OO environment than the 3GL environment.

Field Study Results

The study employed a field study method. Data was collected from three (3) organizations using COBOL and one (1) organization using SmallTalk. 74 data points were collected in the 3GL environment and 17 data points were collected in the OO environment. These 91 data points represent the data for the study.

Data was collected via questionnaire that was to be completed by the programmer upon completion of the maintenance project. Survey questions used a 7-point Likert Scale ranging from 1 (Strongly disagree) to 7 (Strongly agree). Individual questions were grouped to represent a single construct in the model. The constructs represent the various maintenance activities in the model (Planning, Knowledge Building, Diagnosis, and Code Modification). For example, the Diagnosis construct used questions such as:

I frequently compared actual results of a program run to the expected results for that run.

The questionnaire used in the study was published and validated in a previous study (Author).

Analysis of the propositions was done by grouping the constructs into the two activities suggested by the model. Understanding included Planning, Knowledge Building, and Diagnosis. Transformation included Code Modification. A score for each participant on each of the two activities was determined by calculating a mean based on their scores the constructs that make up the activity. Differences between the OO environment and the 3GL environment are tested using MANOVA. The statistical test indicates that there is a significant difference between the activities undertaken in the OO environment and 3GL maintenance environment. (Wilk's Lambda, F=0.877, alpha=.020).

To identify what the differences in the environments might be, and test the propositions concerning differences in maintenance activities, univariate tests were performed on the data. Descriptive statistics are presented in Table 1.

Table 1. Descriptive Statistics (Activities)

	3rd Generation Language		Object-Oriented Language	
Dependent Variable	Mean	Std. Dev.	Mean	Std. Dev.
Understanding	3.627	1.215	4.214	0.697
Transformation	3.446	1.817	4.838	1.172

Table 2. Statistical Test for Differences in Activity Means

Dependent Variable	F	Significance	Eta Squared
Understanding	3.669	.059	.040
Transformation	9.068	.003	.092

Univaritate statistical tests were performed to determine if there is a significant difference in the activities between the environments (Table 2). The results of these tests indicate that there is a statistically different mean in the Transformation activities. There is not a statistically significant difference in Understanding activities. However, it suggests that there may be a difference since the test results indicate a difference at the .10 level. The results indicate that there appears to be a difference in the maintenance activities undertaken in OO and 3GL environments. This suggests that performance measures in the two environments may also differ.

The data was also examined by examining differences between the two environments in terms of the constructs that make up the two activities. Again using MANOVA, the statistical test indicates that there is a significant difference between the activities undertaken in the OO environment and 3GL maintenance environment. (Wilk's Lambda, F=2.270, alpha=.044).

To identify what the differences in the environments might be, and test the propositions concerning differences in maintenance constructs, univariate tests were performed on the data. Descriptive statistics are presented in Table 3. Univaritate statistical tests were performed to determine if there is a significant difference in the activities between the environments (Table 4).

Table 3. Descriptive Statistics (Constructs)

	3rd Generation Language		Object-Oriented Language	
Dependent Variable	Mean	Std. Dev.	Mean	Std. Dev.
Planning	3.595	1.544	4.177	1.064
Knowledge Building	3.296	1.360	3.621	0.712
Diagnosis	3.989	1.767	4.843	1.008
Code Modification	3.446	1.871	4.838	1.172

Table 2. Statistical Test for Differences in Construct Means

Dependent Variable	F	Significance	Eta Squared
Planning	3.680	.058	.040
Knowledge Building	1.715	.194	.019
Diagnosis	8.569	.004	.088
Code Modification	9.068	.003	.092

Closer examination of the differences between the two environments by comparing the construct means provides further evidence that there is a difference in how maintenance is performed in the object-oriented and third-generation programming environments. The statistical tests identified significant differences between the two environments in Diagnosis and Code Modification. The tests also suggested that there may be a significant difference in Planning. For all activities, OO maintenance programmers perceived that they did more of each of these three activities than their counterparts in the 3GL environment. The result is interesting because it suggests that OO programs are more difficult to maintain than 3GL programs. These results are preliminary because of the small number of data points in the OO area and because all the data in the OO area comes from one organization. The second study, in progress, will perform the same analysis to attempt to develop stronger results.

References

References available on request.