Current MIS practitioner literature commonly asserts that commercial 4GLs should completely replace 3GLs in the area of business information processing. One important question in this regard is whether the processing speed of applications written in 4GLs is comparable to that of their 3GL counterparts. This paper examines this central question, presenting the results of a controlled experiment that compared the processing efficiency of FOCUS and COBOL in an IBM PC/AT environment. The experimental results indicated that FOCUS programs, on the average, execute slower compared to their COBOL counterparts. Operating efficiency of applications is considered to be an important contributory factor toward users’ perception of system quality. Thus, given the current state of 4GL technology, the assertion that the commercial 4GLs should completely replace COBOL may be premature. For those applications where efficiency is an important issue, the traditional 3GLs such as COBOL may still be preferred to commercial 4GLs such as FOCUS.
The 4GL system software tends to the details, thus making it much easier and quicker to develop applications. All 4GLs employ very high-level instructions not present in 3GLs. Consequently, with these languages the programmers can create applications with fewer instructions than are needed with 3GLs. Because of this apparent (programmer) productivity gain, it is a common claim in some of the current MIS practitioner literature that commercial 4GLs should completely replace 3GLs (particularly COBOL) in the area of business information processing (Abbey, 1984; Martin, 1985; Read & Harmon, 1981).

From the perspective of MIS management, however, productivity gain in application development is only one concern. Another major concern is the operating efficiency of applications, especially in the context of high-volume, frequently-used production applications. Hence, if 4GLs were to completely replace COBOL, an equally important issue is whether the processing efficiency of applications written in 4GLs is comparable to that of their 3GL counterparts. This paper reports the results of a controlled experiment that addressed this issue.

The organization of the paper is as follows. First, we provide a background discussion of the specific problem addressed in this study. Next, we present the rationale for the choice of languages included in this study. Then, we describe the conduct of the experiment. Finally, we discuss the statistical procedures used to analyze the experimental data and report the experimental results, followed by concluding remarks at the end.

Problem Definition and Research Objective

There exist conflicting claims regarding the processing efficiency of 4GLs vis-a-vis 3GLs. The general notion is that the processing efficiency of 4GL programs is poorer than that of their 3GL counterparts. It is commonly believed that 4GLs are slow and expensive; they generate inefficient executable code, using an excessive amount of computer time and resources (Grant, F., 1985; Kolodziej, 1987b). The usual explanation for this expected inefficiency is that programmers are somewhat more distanced from the computer in a 4GL environment. With 4GLs the programmer has less functional control over task execution by the machine than with 3GLs. 3GL code can be compiled and executed. Most non-procedural 4GLs are executed interpretatively. Compiled code theoretically should execute more efficiently than interpreted code.

As a rebuttal to this general notion, however, many leading 4GL vendors and “experts” contend that such an argument is based on misconception. They argue that it is not necessarily true that (all) 4GLs are less efficient than their 3GL counterparts; there are some comprehensive 4GLs that execute as fast as optimized third-generation code (Coverdale, 1985; Mallery, 1987; Martin & McClure, 1983).

Along these lines, Dan Grant, a specialist on FOCUS, states that the “traditional misconception” of 4GLs being “great but slow and expensive” exists because “numerous applications have been built with no real understanding of internal efficiency or design criteria,” and that 4GLs are “second to none” if used “rightly” (Grant, D., 1985, p. 20). Similarly, Richard Cobb, who was instrumental in the development of RAMIS II, states:

Although the performance of fourth generation languages have (sic.) been found wanting in many instances, it is important to recognize that there is no reason, in principle, why the typical fourth generation language program should be less efficient than its third generation counterpart. In fact there are good reasons why fourth generation software should be more efficient. The primary reason for this contention is that a fourth generation system ‘understands’ the program it processes to a much larger extent than
10 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage: www.igi-global.com/article/experimental-investigation-comparative-operating-efficiency/55658?camid=4v1

This title is available in InfoSci-Journals, InfoSci-Journal Disciplines Computer Science, Security, and Information Technology. Recommend this product to your librarian: www.igi-global.com/e-resources/library-recommendation/?id=2

Related Content

An Extension of the Technology Acceptance Model to Determine the Intention to Use Biometric Devices
www.igi-global.com/chapter/extension-technology-acceptance-model-determine/18153?camid=4v1a

Design and Development of Intelligent Decision Support Prototype System for Social Media Competitive Analysis in Fashion Industry
www.igi-global.com/article/design-and-development-of-intelligent-decision-support-prototype-system-for-social-media-competitive-analysis-in-fashion-industry/148144?camid=4v1a

Information Imbalance in Medical Decision Making: Upsetting the Balance
www.igi-global.com/chapter/information-imbalance-medical-decision-making/18284?camid=4v1a

Studying the Translations of NHSnet
www.igi-global.com/article/studying-translations-nhsnet/3741?camid=4v1a