Abstract

Legacy systems are vital to an organization, and sometimes form the backbone of an organization, yet their maintenance and evolution had been an area of research for a long time. Besides being costly to maintain, legacy systems often lag behind changes in the businesses they support. The challenge in today’s environment is to develop a methodology to migrate older systems to newer, more cost effective client-server distributed processing platforms that support standards-based modular architectures. One approach is to employ a “wrapper” of code that surrounds the existing legacy code, turning it into an object. This could be stated as an object oriented approach to legacy systems. However, there are many other paradigms that a legacy system might adopt. Aspect-oriented technology is another emerging programming paradigm that is receiving considerable attention from research and practitioner communities alike. Nowadays much of the work is carried on, on developing different methodologies to enable aspect oriented programming to be applied to legacy systems. In this paper, we try to analyze the impact of object oriented technology and aspect oriented technology on legacy systems and the environment that is required to implement the two paradigms. The advantages and disadvantages of both the paradigms have been explored, and a comparative study of both the paradigms is done and analyzed in the light of legacy systems.
Reference

- Jan Hannemann, "Aspect-Oriented Refactoring: Classification and Challenges", 2006
- Jaime Gómez, Cristina Cachero, and Antonio Párraga, "Extending UML for the migration of Legacy Systems to the Web", Spain, 2002
- Peter Ebraert and Tom Tourwe, "A Reflective Approach to Dynamic Software Evolution", 2004
- Kris De Schutter, Bram Adams, "Face-off: AOP+LMP vs. legacy software", 2007
- Nader Mohamed and Jameela Al-Jaroodi and, "An Object-Oriented Approach for High Availability of Applications Integration", United Arab Emirates University, 2007
- UIML http://www.uiml.org
- www.rational.com/uml/index.jsp

Index Terms

Computer Science Programming Languages

Key words

Legacy Systems
Aspect Oriented Programming
Refactoring techniques
Object Oriented Refactorings

Procedural languages challenges