International Journal of Computer Science
and Engineering (IJCSE)
ISSN(P): 2278-9960; ISSN(E): 2278-9979
Vol. 3, Issue 1, Jan 2014, 105-112

International Academy of Science,
Engineering and Technology
Connecting Researchers; Nurturing Innovations

EXPANDING ASPECT-ORIENTED PROGRAMMING TOWARD ADVANCED DYNAMIC NETWORK LEVEL ASPECTS IN C++

RAWAN KASASBEH, SUFYAN AL-MAJALI & ARAFAT AWAJAN

Department of Computer Science, Princess Sumaya University for Technology, Amman, Jordan

ABSTRACT

Programming models are continuously improving to achieve two principals, more modularity and less crosscutting concerns. The final improvement was the emergence of Aspect Oriented programming (AOP), AOP emerged to improve previous programming models rather than replace their work. The main improvement AOP has caused was the increase in program modularity, reduction of code redundancy and less code scattering. This is done through gathering the scattered code in one separate class called an aspect, weaving this aspect where needed according to point cuts, this way the redundant code is limited which results in better code modularization.

Our approach proposes the use of AOP along with networking, through allowing the introduction of a new aspect at runtime not only on a single computer, but at a network level. Therefore, there are two dimensions to our work, first allowing the aspect to be introduced at runtime gives better dynamicity and availability to the application, the program is dynamic since it is applied at runtime of the program and does not need to be introduced at runtime, it is available since there is no need to shutdown or restart the program in order to introduce the new code. The second dimension is adding this work at a network level.

Having a client server like network and having the option to add an aspect to the specified locations on some or all the client computers, weaving or unweaving the aspect at runtime. As mentioned this work increases dynamicity, flexibility and availability on the expense of system performance. Although the user should prioritize the most important requirements before using ADAC++ whether it is more dynamicity and availably or better performance.

KEYWORDS: Aspect Oriented Programming, Advanced Dynamic Weaving, Runtime Weaving, Weaving at Network Level Dynamic Weaving