Wishes for object-oriented languages

Markku Sakkinen

Department of Computer Science and Information Systems
P.O.Box 35 (Agora)
FIN-40014 University of Jyväskylä
Finland
sakkinen@cs.jyu.fi

ABSTRACT. Today, the emphasis of research and development in object-oriented software engineering has moved from classes, methods and other base-level entities to components, frameworks and other larger entities. Consequently, interests have also moved from programming languages to environments and tools for manipulating these entities. This evolution is natural and welcome, but it does not mean that the existing object-oriented programming languages are already perfect and no further progress on that level is needed. I will present several things that I would wish to be improved in current languages or their successors. Most of the points in the talk are not completely new, but some may be little known or forgotten. Several of my wishes are also controversial, and I will be happy if they stimulate discussion. A major part of the paper will discuss inheritance, which is the most conspicuous characteristic of object-oriented languages. I still believe that even multiple inheritance can be done right, but do not know of any existing language in which it has been done right. I will propose one-to-one correspondence between ancestor and heir subobjects as perhaps the most essential requirement for avoiding multiple-inheritance anomalies. Current languages have defects also in some other aspects where object orientation has often been advertised to be good and natural. Among these, I will discuss at least the handling of composite (or complex) objects. A timely topic is genericity, which has at last been added also to Java; in this aspect, Ada still seems to remain ahead of most competitors. Some other things have already been analysed by better experts, such as the problems of the concurrency principles of Java by Per Brinch Hansen.

KEYWORDS: object-oriented languages, inheritance, composite objects, genericity.