Object-Oriented COBOL: The Old, The Bad and The Ugly?

Guido Dedene
Catholic University of Leuven, Belgium

This tutorial will present a critical appraisal of the current standards proposals for COBOL, including object-orientation. First of all, the fundamental characteristics of object-orientation will be mapped on the new COBOL standard, with special attention for the principles of encapsulation and uniform access. The discussion of the characteristics allows to position the COBOL-vendors in terms of the object-orientation perspective they have implemented in their technology. In particular, the constant struggle between weak and strong typing can clearly be illustrated. The tutorial will also address the integration of object-oriented COBOL with new and existing systems. In particular, wrapping techniques for transaction and DBMS-based systems are discussed, and an example is presented on how to integrate COBOL-code in Java and Eiffel software, through the appropriate use of C-generation and the Java Native Interface(JNI). Moreover, the integration of object-oriented COBOL with component brokering technology (CORBA versus (D)COM(+) versus JAVA RMI) is discussed. The tutorial will also discuss how the principles of "Design by Contract" can be implemented in object-oriented Cobol. The tutorial includes the presentation of a fully elaborated UML-case, with the full details of alternative object-oriented Cobol implementations. The tutorial will also emphasize the integration of elements of structured programming (based on JSP) with the object-oriented features of the new Cobol standard. Finally, the integration alternatives for the integration of legacy systems and/or commercial off the shelf (COTS) components is discussed, and illustrated in detail. The tutorial will present examples in various formats, including the technologies of MERANT (MicroFocus), IBM and Fujitsu. Some experiences with object-oriented Cobol in real case studies will be presented. Finally, it will be shown how Eiffel can be used as a didactical instrument to teach and train the object-oriented principles, and how they can be reinforced in object-oriented Cobol implementations afterwards.

Tutorial Agenda

- Introduction: the COBOL saga
- Introducing Object-orientation in COBOL: classes and methods in COBOL.
- Discussion of OO-aspects in OO-Cobol: strong versus weak typing, encapsulation, abstract classes, Inheritance, Design by Contract
- Elements of structured programming, integrated in OO-Cobol.
- Discussion of an elaborated case study; from UML to OO-Cobol code
- The case study is a small library application, including reservations.
- Discussion of object-oriented standards versus Cobol, in particular integration of (legacy) Cobol code with object-oriented software.
- Discussion of Cobol resources available on the world-wide web.
- How to teach OO-Cobol (using principles from the Eiffel language/method)
- The future of Cobol

Audience background

A broad audience is targeted with this tutorial. All concepts of the object-orientation as well as Cobol are briefly explained in the notes and slides for this tutorial. Project managers will be able to assess the differences between the different Cobol implementations and the headlines of the new Cobol standards proposals. It is assumed that the participants have a basic understand of Cobol, not necessarily from a Cobol programming point of view (all basic Cobol concepts needed for the tutorial are included in the notes).
Benefits of participating
The participant will enjoy the following benefits for attending this tutorial:

• A positioning of the proposed COBOL (2002?) standard, in confrontation with the principles of structured and object-oriented design and programming.
• An understanding of how object-oriented Cobol may contribute to the smooth introduction of object-oriented concepts in a heavily legacy-code-biased software development group.
• An insight in the subtle accents in the proposed COBOL standard, and the way how the vendors implemented these accents. In particular, the influence of strong versus weak typing, and its consequences in distributed systems are discussed in detail. The fact that the presenter is academic in nature contributes to the neutral presentation of the facts!
• An understanding of the standardization process and the difficulties related to it. In particular, an assessment on why the COBOL97 standard was indeed premature.
• Participants will assess a full life-cycle case study, starting from an UML-based analysis and design up to a final implementation in object-oriented Cobol.

Guido Dedene holds a phd. in Mathematics and is full professor of Business Information Systems at the Faculty of Economics and Applied Economics of the Katholieke Universiteit Leuven (K.U.Leuven, Belgium). His teaching and research areas include software development, economics of information systems and computer performance, and higher management education on information systems. Prof. Dr. Guido Dedene is co-founder of the Leuven Institute for Research on Information Systems (LIRIS) at K.U.Leuven and researcher at the PRIMAVERA institute and the A. Dreesman Institute for Infopreneurship at the University of Amsterdam. He is also professor on "Information and Communication Systems" at the University of Amsterdam. He holds board positions at the Library Value-added network LIBIS-Net, the E-Library Software company ELiAS and the Information and Communication Technology and Systems Strategy Council of the K.U.Leuven. Guido Dedene was very active in Guide and Share Europe since 1986, first as Region Manager for the Belgium-Luxembourg Region, next as European Division Manager for Application Development and finally as Director for European Services until 1998. He conducted several projects, including the Application Development Joint Project ADJP, and holds several conference speaker and publication awards. He published in refereed journals, including IEEE and ACM, and received the IEEE Software Best Practice Award in 1995.