ICIS 1993 Proceedings

International Conference on Information Systems (ICIS)

1993

TUTORIAL 2 AN INTRODUCTION TO THE OBJECT PARADIGM AND OBJECT-ORIENTED SYSTEMS

David E. Monarchi
University of Colorado, Boulder

TUTORIAL 2

AN INTRODUCTION TO THE OBJECT PARADIGM AND OBJECT-ORIENTED SYSTEMS

David E. Monarchi
University of Colorado, Boulder

"Object-oriented" is one of the key phrases of the decade. The object paradigm has been variously billed as everything from "a great revolution, how we will build the information systems of the future" to "an evolution, not a revolution" to "much ado about nothing." Companies are admonished to go OO or be left behind as their competitors gain an advantage by building OO systems. The tutorial will provide a basic understanding of the object paradigm and of object-oriented (OO) systems for those with little or no familiarity with them.

The tutorial will begin with a small example to introduce the principle concepts of the paradigm in a specific setting, and to give a sense of what an OO system is and how it "runs." The concepts will include abstraction; classes, instances, and objects; methods, messages, and polymorphism; generalization, specialization, and inheritance; aggregation; (other) associations; and encapsulation and data hiding. As we examine the implementation of the example, we will be looking for answers to the questions "Where is the program? How does it work? How does it start?" We will see also some of the differences between a structured program and an OO one.

Next we will examine the process of creating an OO system. The OO life cycle is generally considered to be a prototyping-like process involving repeated cycles through modeling and implementation. Modeling involves OO analysis and logical design; implementation involves physical design and OO programming. The principle boundary in the process is between modeling and implementation. Two of the alleged strengths of the paradigm are the blurring between analysis and logical design in modeling and the reduced gap (compared to traditional structured approaches) between modeling and implementation. We will look at a framework for classifying the various modeling tools (e.g., object diagrams and state transition diagrams) with some examples from our earlier problem.

Of course, OO systems don't build themselves. So next we will spend a few minutes discussing the management of OO projects because the best technology and ideas won't help much if you can't get them into practice.

"Reuse" has become to some people the key concept and promise of the object paradigm. Reuse is supposed to reduce development time, eliminate bugs, improve quality, etc. We will briefly examine some types of reuse and offer some comments on these claims.

Finally, as you might expect, a topic as "hot" as the object paradigm is generating a great deal of research, both applied and theoretical. Time permitting, we will identify some current key research areas.

A list of definitions and a selected reading list will be available at the tutorial. The example system will be available from the speaker.