Chapter I

Object Oriented Requirements Analysis: Its Challenges and Use

Edward R. Sim
Loyola College, USA

Introduction

The ability to correctly identify system requirements is seen by most Information Systems (IS) researchers and practitioners as essential to the design and development of effective information systems (Yadav, Bravoco et al. 1988; Vessey 1994). Requirements are used to drive all subsequent stages of systems development and are critical to system validation. Incorrect requirements or poorly specified requirements usually produce systems that require major revisions or are abandoned entirely (Pressman 1996). Recently, many new techniques and methodologies have been introduced to assist analysts and users in efforts to identify and specify system requirements (Coad, North et al. 1995) (Pancake 1995). One of the newest approaches to be used in this effort to improve requirements analysis is the application of object oriented analysis (OOA).

Proponents of OO argue that the use of OO concepts improves communication and the formation of accurate conceptual models because OO’s fundamental concepts are more “natural” (Booch 1991). They argue that the use of OO facilitates communication and problem understanding because people naturally think about their environment in object oriented ways (Martin and Odell 1992). The solution for
improving analysis and the requirements produced by that analysis, they argue, is to adopt an OO approach to doing analysis. However, despite these claims, the use of OOA has not achieved the levels of adoption that other object oriented technologies (i.e., programming languages) have achieved. This chapter examines OOA, reviews some of the fundamental concepts on which OOA is based and discusses the acceptance of this new technology.

**OO Requirements Analysis**

Requirements drive the function to structure transformation that occurs during IS development. The requirements for a particular IS development effort represent the goals or tasks that system must meet in order to be successful (Davis 1993). Most often these goals or requirements are specified as a set of functions and constraints that the system must meet (Yadav, Bravoco et al. 1988). Usually these requirements are expressed at a relatively high level of abstraction (i.e., the system must provide customer purchase information) and are later refined to detailed specifications.

Many researchers and developers of IS methodologies divide the requirements activities into discrete stages: problem analysis and description (Norman 1988). During the problem analysis the analyst seeks to understand the problem by identifying essential problem elements and structuring those elements into a coherent problem description. The problem description becomes the basis on which a solution is proposed and subsequent specifications are written. These two major activities of requirements are referred to as requirements elicitation and requirements specification (Whitten, Bentley et al. 1994). Various techniques for performing the activities of each of these stages have been proposed. In most cases the techniques suggested by the particular development methodology attempt to guide the user in constructing various graphical models (sometimes augmented by textual descriptions) that describe the current and future systems. The requirements output consists of a set of models and/or textual descriptions produced by following the techniques and heuristics suggested by the methodology.

The specific output products of the requirements activities vary according to the IS development methodology being followed. In most cases, structured analysis leads to the development of data flow diagrams, textual process specifications and a high level data dictio-
Related Content

**From Bad Smells to Refactoring: Metrics Smoothing the Way**  
[www.igi-global.com/chapter/bad-smells-refactoring/27362?camid=4v1a](www.igi-global.com/chapter/bad-smells-refactoring/27362?camid=4v1a)

**Object Oriented Testing in Software Development**  
[www.igi-global.com/chapter/object-oriented-testing-software-development/27346?camid=4v1a](www.igi-global.com/chapter/object-oriented-testing-software-development/27346?camid=4v1a)

**RUP and eXtreme Programming: Complementing Processes**  
[www.igi-global.com/chapter/rup-extreme-programming/25636?camid=4v1a](www.igi-global.com/chapter/rup-extreme-programming/25636?camid=4v1a)
Web Services in Service-Oriented Architectures
www.igi-global.com/chapter/web-services-service-oriented-architectures/25640?camid=4v1a