Chapter XI
Performance Case Modeling

Ian Douglas
Florida State University, USA

ABSTRACT

This chapter introduces performance case modeling as a means of conducting a performance analysis. It argues that the design of any instruction focused on practical subjects should be preceded by understanding of the performance requirements for graduates of a course of instruction. This understanding is facilitated by the collaborative creation of diagrams that identify the different roles a performer takes and their associated goals, together with documentation of performance measures for the goals. The measures serve as a baseline for the evaluation of instructional effectiveness. Other approaches to visual languages in instructional design have been more focused on modeling the architecture of the instructional system rather than the performance environment in which its graduates will be expected to perform. The approach described is based on UML use cases and serves to focus thinking on the performance analysis that should occur prior to the design of instruction.

INTRODUCTION

The performance analysis stage of most instructional system design process models has long been seen as one of the most crucial stages (Harless, 1970). If designers do not sufficiently understand the problem they are unlikely to create an optimal solution. In addition to aiding the understanding of the problem, performance analysis is also focused on establishing performance measures and collecting initial data. It is not possible to determine the value of a solution, such as a new course of instruction, without having analysis data that would allow one to show improvement over a baseline level of performance (Deming, 1982). This is especially true if designers are aiming to achieve Kirkpatrick's level 3 (on the job application of acquired skills and knowledge) and level 4 (improved organizational outcomes as a result of the acquired skills and knowledge) (Kirkpatrick, 1998). Clark and Estes (2002) present a number of case studies illustrating that data-driven analysis leads to better solutions for performance improvement.

Some terms that are often used interchangeably in performance analysis are: needs assess-

ment (Kaufman, 1988), needs analysis (Mager & Pipe, 1984), performance assessment (Robinson and Robinson, 1995), front-end analysis (FEA) (Harless, 1988), and training needs analysis (Rossett, 1998). In this chapter the term performance analysis is used as a general term for all of these types of analyses.

The majority of attempts to adapt visual languages for instructional design have been focused on designing the solution space (the nature of the instructional system). In this chapter we will consider an approach that is specifically focused on understanding the problem space (the performance requirements for the employers of the instructional system graduates). Understanding the desired performance is an important first step that should occur prior to developing instruction or any other means of improving human performance.

The specific approach is an adaptation of use case modeling, which is part of the unified modeling language (UML). UML is a systems modeling tool which, although developed primarily for computer systems modeling, is adaptable to the modeling of other types of systems. Despite its origins in software engineering, elements of it have been adapted for such diverse purposes as business process modeling (Marshall, 2000; Eriksson and Penker, 2000) and educational modeling (IMS, 2005). Different aspects and applications of UML relative to instructional design are described in a number of chapters in this book. In this chapter we will focus on one area of UML that has been relatively neglected.

UML is not a single language but rather a collection of diagramming techniques and specification languages that serve different purposes in systems analysis and design. It can be used both to model existing systems and to envisage models of new systems. The nine different diagramming techniques found in UML serve different purposes: describing what the static structure of a system is like (class, object, and component diagrams), describing how a system operates and information flows through the system (activity, collaboration, sequence, and statechart diagrams), describing how a system is deployed (deployment diagram), and describing the functional requirements of the system (use case diagrams). A number of the chapters in this book describe the adaptation of elements of UML for instructional systems architecture design (e.g., Chapter IX). They primarily focus on the use of Structure and Flow diagrams. In this chapter the focus will be exclusively on use case diagrams.

Given that they are concerned with system inputs and outputs rather than the internal technical details, use cases are often seen as a crucial part of UML when used in software development. Use cases aim to establish what an efficient and effective system should achieve. They serve as the starting point and driving force behind all other analysis and design activities (Jacobson, 1992; Rosenberg & Scott, 1999).

This chapter explains the concept of use case modeling, shows how it can be adapted for performance analysis (performance case modeling), discusses software tools for repositories and reuse, and suggests guidelines for applying performance case modeling.

PERFORMANCE ANALYSIS

There are a number of approaches to performance analysis prescribed and studied in the literature (Rummler & Brache, 1995; Swanson, 1994;Wedman & Graham 1992; Schaffer, 2000).

The major tasks of performance analysis that they tend to share are the following process:

- Identify what people in particular roles are required to achieve.
- Identify gaps between exemplary (or expert) performers and typical performance or novice performers.
- Analyze causes for those gaps.
- Identify and select the solutions (e.g., required instruction) to close the gaps.