Chapter XVII
The Pervasiveness of Design Drawing in ID[1]

S. Todd Stubbs
Brigham Young University, USA

Andrew S. Gibbons
Brigham Young University, USA

ABSTRACT

This chapter is a survey of the literature of ID to look at the breadth and usage of design drawings in this discipline to better understand the emerging use of VIDLs to improve designs. To conduct this research, we sampled several ID textbooks, ID journals, software, and case studies looking for examples of design drawing. Design drawings found were then categorized using Gibbons' (2003) seven ID layers as a taxonomy to understand the drawings' purposes. We did not find the same pervasiveness or level of self-awareness as found in other design fields. Examples of design drawings were found, but were somewhat rare. Furthermore, we discovered that those examples we found tended to document only two of Gibbons' seven layers, indicating narrow application. We believe this gap represents a serious shortcoming in ID, indicating a lack of tradition, skill, and standards for visual representations of design except in limited ways. At present, design drawing is a rare but growing phenomenon in ID, which, when fully understood and implemented, can only benefit the practice of ID.

INTRODUCTION

This chapter applies a layered concept of instructional design (ID) architecture described by Gibbons and Rogers (in press) to a taxonomy of design drawings described by Stubbs (2006) to produce a refined category system for describing the use of drawing and sketching in ID. The value of doing so is dramatized by Stubbs, who compares the use of design drawing in ID to its use in other design fields, detecting a large disparity. If Stubbs' analysis is correct, then designers in other fields have a much richer tradition of the use of drawing in design and a literature that shows a much higher level of self-awareness in the use of drawings during design than most instructional designers would expect.

Design drawing might be considered the primitive of visual instructional design languages (VIDLs). In this chapter we hope to understand where we are with this basic form of VIDL to better understand where we are going.

Though instructional designers excel in the use of drawings of many kinds in their *produced* designs, it would appear that they lag behind other design fields in exploiting the value of drawings and sketches while *designing*.

This deficit has important consequences for the economics, quality, and quantity of instructional designs. Whereas other design fields have begun to capitalize on the power of computers as a design tool, instructional designers seem to be more at the mercy of the tools and design interfaces created for use by others who have more vibrant economies, such as Web and software design. Early attempts to create tools to express designs in the instructional designer's vernaculars appear to have been swallowed up in the success of other design fields, notably the Web and Web development tools (Fairweather & Gibbons, 2000). Only recently has interest in the authoring of learning objects revitalized interest in design interfaces that emphasize ID structures, a trend that we hope will persist and broaden.

The value of the computer to design lies in its ability to take part in routine and mundane decision-making. Successes in computer-aided design have come largely from the ability to describe a design problem (or some portion of a design problem) in terms that can be translated into computer languages. For instance, the design of an architectural column can be translated into sub-problems for the design of the capital, the shaft, and the column base (Mitchell, 1990). If only the shaft sub-problem could be expressed in computer terms, then that portion of the design could be given computer support, and the remaining sub-problems would depend entirely on human decision-making. By the same reasoning, if only portions (sub-problems of sub-problems) of the design of each of the capital, shaft, and column base could be expressed in such terms, then the design of each of these would require human effort and decision-making, supplemented by some degree of computer assistance. This is the principle today of popular development systems for Web and software design. The involvement of the computer—which is capable of making numerous routine decisions very rapidly and dealing with representation issues at the same time—creates an economic lever. More quantity at higher quality can be produced more rapidly—cheaper, better, faster. And as languages for problem description and solution improve and become more nuanced, the quality and sophistication of the designs improves. This is exactly what has happened to the design of computer chips over the past thirty-five years. Chips designs today are created to human specifications with human decision-making concentrating mainly on high-level design issues. As a result the economics of computer chip design have changed so that a return to hand-drawn circuit design would be an expensive luxury.

This chapter addresses how ID problems can be described in terms of design languages (some portion of which may be translatable into computer languages). It begins by describing research by Stubbs (2006) on the use of design drawing by