Learner-Centered Design: Developing Software that Scaffolds Learning

Chris Quintana and Elliot Soloway
Center for Highly Interactive Computing in Education
University of Michigan
quintana@umich.edu

Cathleen Norris
Department of Technology and Cognition
University of North Texas

Abstract

Learner-centered design (LCD) is an evolving design approach for designing tools that support learners trying to engage in and understand complex work practices in which they are novices. Learner-centered design focuses on developing tools that incorporate support (or “scaffolding”) to support the novice in seeing and doing complex, unknown work so that the learner can begin developing an understanding of that work in a “learning by doing” fashion. In this tutorial, we will give participants a more in-depth treatment of learner-centered design by describing a definition for LCD, theoretical background, and the LCD process and methods to give participants experience with the mindset needed for the LCD approach.

1. Introduction

Learner-centered design (LCD) is an evolving design approach addressing the needs of learners—a specific audience trying to work in and understand a work practice in which they are novices [6]. The central tenet of learner-centered design involves the design of scaffolding into software to make complex work activity accessible to novices [6]. Scaffolding allows learners to engage in unknown work activities that would normally be out of their reach to begin developing an understanding of the work [7].

Many researchers are exploring learner-centered design as evidenced, for example, by two special issues of Communications of the ACM focusing on LCD, a workshop at the ACM CHI 99 human-computer interaction conference, and previous tutorials at CHI and other university settings. Here, we continue this work with a more in-depth treatment of learner-centered design by describing a LCD definition, theoretical background, and the LCD process and methods to give participants experience with the mindset needed for LCD.

There are two basic, high-level learning objectives for this tutorial:

• Objective 1: Learning what learner-centered design is. Participants will learn background about LCD: LCD definitions and descriptions, theoretical background, and examples of learner-centered tools.

• Objective 2: Learning how to “do” learner-centered design. Participants will learn design approaches for doing learner-centered design. Through examples presented in the tutorial, participants will see design methods for designing learner-centered software.

2. Objective 1: Defining Learner-Centered Design

We will describe several high-level themes about LCD:

• “Learning by doing” approach. Learner-centered design is informed by a “learning by doing” approach that says learners gain expertise in new work by mindfully performing authentic activities from the work practice.

• Scaffolding in software. Learner-centered software incorporates “scaffolding” to support learners as they do new work. Scaffolds can be considered to be software features that support work practice novices in performing previously inaccessible work activity. As these novices develop expertise in their work, the scaffolds are no longer needed and can fade away from the software.

We will also give a more structured definition of learner-centered design by comparing LCD to user-centered design (UCD) [1]. Specifically, we will compare UCD and LCD along three dimensions [3]:

• The audience being addressed by each design perspective.

• The central design problem being addressed by each design perspective.

• The underlying theory used by each design perspective to address the design problem.

Audience. We will discuss the audience addressed by each design perspective by taking into consideration the audience’s level of work expertise, growth, diversity, and motivation. In UCD, the audience of users is considered knowledgeable and motivated about their work. The user audience is less diverse because designers design around their tasks, not the users themselves. Finally, users grow less in their work expertise.

In LCD, the audience of learners—primarily K-16—has little and inaccurate work expertise and wavering motivation to engage in the work. Learner diversity is a key issue because of the differences in gender, learning
styles, etc. Finally, learners should gain work expertise, so the growth of the learner must be considered in the design of the tool.

Central Design Problem. We will discuss the central design problem being addressed by each perspective. In UCD, the central design challenge is to design tools that are easy to use and understand and that allow the easy and efficient completion of work activities. Because tool use is the central challenge, designers must address the conceptual gap between user and tool (i.e., the gulf of execution and evaluation).

In LCD, the central design challenge is to design tools to help learn new work practices. Because learning new work practices is the central challenge, designers must address the conceptual gap between learner and work of expertise (i.e., a gulf of expertise).

Underlying Theory. We will discuss the underlying theory used by each design perspective to address the central design problem. In UCD, the underlying theory is a “theory of action” that describes how people use tools. In LCD, the underlying theory includes social constructivist learning theories that give a perspective on how people learn.

3. Objective 2: Describing the Learner-Centered Design Process

Given a description of learner-centered design, we will describe some design methods and techniques that can be used to develop and evaluate learner-centered software. The traditional approach to software design involves high-level phases such as task/work analysis, requirements specification, design, and evaluation.

We will describe how we should modify the traditional software design cycle to develop specific methods within each phase for learner-centered design. Our primary example for doing so focuses on the design of Symphony, a tool to support the work of science inquiry [4]. Specifically, we will describe:

- Work analysis: How can designers study the work practice to fully understand, especially understanding the tacit knowledge that experts use to perform the work? We will discuss the process-space analysis method for analyzing work practices to isolate different components of the work that need to be made explicit and accessible to learners [4].
- Requirements specification: What are learner-centered software requirements? Identifying learner-centered requirements involves identifying learner support needs that isolate the areas where learners will need support to engage in the new work practice.
- Design: What kinds of conceptual and physical user interface and software design decisions are needed for LCD? Designers first need to identify the scaffolding strategies that will address the learner support needs identified earlier. Second, designers need to determine how those scaffold strategies will be implemented in software to help learner mindfully engage in the new work practice. We ground these ideas with software examples from University of Michigan, Northwestern University, University of California at Berkeley, and Vanderbilt University.

- Evaluation: How can designers evaluate the effectiveness of learner-centered software? Designers need to evaluate scaffolded software to analyze the “effects of” and “effects with” software [5]. Designers need to assess the impact that the individual scaffolds have on the learner’s work understanding (i.e., evaluate the “effects with” software). Designers also need to assess whether learners learned the new work (i.e., evaluate the “effects of” software) [2].

5. Acknowledgements

This material is based on work supported by the National Science Foundation under Grant No. REC 99-80055.

6. References