IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITB8278

Chapter II

Evolutionary Growth and Control in User Tailorable Systems

Anders I. Mørch
InterMedia, University of Oslo, Norway

This chapter is about evolutionary application development as a means for including end users in further development of existing software systems. The chapter presents concepts and techniques for evolutionary development that reuses ideas from other domains in which evolution plays a dominant role (biology, architecture, and art objects). It argues that end users need access to representations of a system that are less formal than program code, but more powerful than informal design representations, and that this information can be obtained from the system's past use and development history. The "resemblance relation" is presented as a tentative solution. It includes elements of object-oriented programming and component-based development. It is hoped that this chapter will provide the reader with a new view on systems development, and how end users can participate as designers in the evolutionary process.

INTRODUCTION

Evolution is about the history of choices leading to a current system or state of development as well as a system's ability to adapt to future requirements. The former aspect is seldom reflected in information systems development, even though the term evolution is frequently used in conjunction with system development activities, such as prototyping (e.g., Budde et al., 1992). This chapter is about the role of user participation in evolutionary application development.[1] Application users are seldom computer experts and cannot be expected to participate in systems development by writing program code. They need access to representations of a system that are less formal than the system itself. A claim is made in this chapter that the kind of information users need access to can partly be obtained from the system's past use and development history (e.g., Ehn, 1988).

I make extensive use of metaphor and analogy (from many domains) to explain basic processes in evolutionary development and to suggest tools and techniques for application systems development. A central metaphor is "user as gardener." Gantt and Nardi (1992) and Christiansen (1997) have introduced this metaphor to suggest new user roles in end-user computing. Their claim is that end users can aid systems development in the same way a gardener nurtures trees and plants in a garden. Two small examples illustrate the chapter's perspective on evolutionary development by identifying some similarities and differences between evolutionary growth in a garden and in an information system, respectively.

A fruit tree is an evolutionary system that grows at two ends (buds and roots). The growth process that starts with a bud can roughly be described as follows: A bud grows to become a small branch, which gives rise to new buds. A branch may later become an inner branch (having itself spawned new branches) and the first branch becomes the stem of the tree, etc. A unique feature of this process is that the tree's visible parts (bud, leaf, branch, stem, etc.) express themselves differently at different stages in the tree's lifetime. This characteristic is found in most living systems but is not commonly observed in information systems development.

Information systems also evolve, but differently. A file system will serve as example. It can be defined as a collection of nodes in a part/whole (composition) hierarchy. Each node can be either a container (directory) or a part (a file). A small file system may consist of one directory and three files. When adding a new file to the system that is related to one of the old files, one option is to create a subdirectory inside the first directory and put the two related files there. When this is done the system has a new organisation that has replaced the old. We get a "reminder" of the old organisation only if the subdirectory has a name that resembles the name of the file that was the source of the subdirectory. However, most information about the past organisation of the system is lost.[2]