SimSE: An Educational Simulation Game for Teaching the Software Engineering Process

Emily Oh Navarro
School of Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425 USA
1-949-824-3100
emilyo@ics.uci.edu

André van der Hoek
School of Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425 USA
1-949-824-6326
andre@ics.uci.edu

Categories and Subject Descriptors
K.3.2 [Computers and Education]: Computer and Information Science Education – *Computer science education*.

General Terms
Management, Human Factors

Keywords
Software engineering education; Educational games

1. INTRODUCTION
A large difference exists between the software engineering skills taught at a typical university and the skills that are desired of a software engineer by a typical software development organization. This problem seems to stem from the way software engineering is typically introduced to students: general theory is presented in lectures and put into (limited) practice in an associated class project. Although both lectures and projects are essential, they lack a practical, in-depth treatment of the overall *process* of software engineering—lectures allow only passive learning, and the size and scope of class projects are too constrained by the academic setting to exhibit many of the fundamental aspects of real-world software engineering processes. To address this problem, we are developing SimSE, a computer-based environment that allows the creation and simulation of software engineering processes. SimSE allows students to "virtually" participate in a realistic software engineering process that involves real-world components not present in typical class projects, such as teams of people, large scale projects, critical decision-making, personnel issues, budgets, planning, and random, unexpected events.

2. SIMSE
SimSE is a single-player game in which the player takes on the role of project manager of a team of developers. The player must manage these employees to complete a particular (aspect of a) software engineering project. Management activities include, among others, hiring and firing, assigning tasks, monitoring progress, and purchasing tools. Generally, following good software engineering practices will lead to positive results while ignoring these practices will lead to failure in completing the project. The user interface of SimSE is fully graphical, displaying a "virtual" office in which the software engineering process is taking place, including typical office surroundings (e.g., desks, chairs, computers), employees, customers, and project information (e.g., budget and time), as well as representations of software engineering artifacts that include such information as that artifact's completeness, correctness, and other similar qualities. Information about the status of individuals is provided through automatic pop-up "bubbles" over the heads of individuals and through explicitly querying an individual. Players use information gleaned from these sources to make decisions and take actions, driving the software engineering process to complete a project within budget, schedule, and at or above the customer's desired quality requirement.

Because many different schools of thought exist about software engineering, and the educational needs and objectives of different instructors vary, the models of the software process that execute within SimSE must be customizable. Therefore, an integral part of SimSE is a model builder that an instructor can use to specify the particular software engineering process that he or she wishes to teach, including: 1) major entities in the simulation; 2) activities that these entities can participate in; 3) rules that specify the effects of these actions on the rest of the simulation; 4) graphical representations of all major entities; and 5) the entities that the game is to start with, or the start state. A customized simulation that the students can play is then generated.

The simulation loop of SimSE operates in the following manner: An engine drives the simulation by emitting clock ticks. At each tick, a rule execution component executes the rules associated with the currently active actions, and then propagates the effects of these actions on the current game state. Following this, the engine signals the user interface to update the display.

To date, the SimSE model builder and code generator have been completed, and development of the graphical user interface and an initial simulation model are both nearing completion. In the near future, we plan to evaluate the teaching potential of SimSE by conducting experiments involving undergraduate computer science students at UC Irvine. By the time ITiCSE takes place, we expect to not only bring a poster describing the principles and underlying technology of SimSE, but to also bring results from the initial experiments, along with a fully working demo of SimSE.

ITICSE'04, June 28–30, 2004, Leeds, United Kingdom.
ACM 1-58113-836-9/04/0006.