INTRODUCTION TO OBJECT-ORIENTED DISCRETE
EVENT SIMULATION WITH THE OOSIML LANGUAGE

José M. Garrido
Department of Computer Science and
Information Systems, Kennesaw State
University
1000 Chastain Road
Kennesaw, GA 30144
770-499-3652
jgarrido@kennesaw.edu

ABSTRACT
This tutorial provides an overview of object-oriented modeling
principles and an introduction to developing object-oriented
simulation models using the OOSimL. The focus is on early
introduction to simulation in the computing curriculum. This
approach to simulation emphasizes and exploits object-oriented
modeling and programming in the large to develop models of
simple to complex systems.

The tutorial presents a gentle introduction to object-oriented
modeling and simulation using the new language OOSimL. This
simulation language was designed and developed for teaching OO
simulation early in undergraduate programs in computing. The
OOSimL compiler generates Java source code (and recently
expanded to generate C++ code). Object-oriented concepts are
reinforced using OOSimL and the simulation approach follows
the process style of simulation.

Categories and Subject Descriptors
I.6 [Simulation and Modeling]: Object-orientation. D.2.2 [Design
Tools and Techniques]: Object oriented design methods. D.3.3
[Programming Languages]: Language Contracts and Features –
abstract data types, polymorphism, control structures.

General Terms
Performance, Design, Reliability, Experimentation, Security,
Languages, Theory, Verification.

Keywords
Process interaction. Active and passive objects. Synchronization.
Resources. Queues.

Permission to make digital or hard copies of all or part of this work
for personal or classroom use is granted without fee provided that
copies are not made or distributed for profit or commercial
advantage and that copies bear this notice and the full citation
on the first page. To copy otherwise, to republish, to post on
servers or to redistribute to lists, requires prior specific
permission and/or a fee.

ACMSE '10, April 15-17, 2010, Oxford, MS, USA.
Copyright © 2010 ACM 978-1-4503-0064-3/10/04...
$10.00

1. INTRODUCTION
The general goals of this tutorial are:
• Increase awareness of simulation in the computing
curriculum
• To provide further application of object-oriented
programming and modeling
• Present a general high-level object-oriented programming
and simulation language for implementing simulation models

Participants to this tutorial will become familiar with the process
style of discrete-event simulation using an object-oriented
programming and simulation language.

The objective of the tutorial is to present the general concepts and
techniques used for modeling and implementing simulation
models using OOSimL. Provide a gentle introduction to object-
oriented programming, modeling, and simulation. Discussion on
implementing undergraduate OO simulation courses in computing
and to help strengthen the application areas of computer science.

2. FORMAT USED
The audience for this tutorial is: a) faculty teaching programming,
OO modeling and design, and/or simulation, b) students interested
in the mentioned areas, and c) practitioners of OO modeling.

The duration of tutorial: 75-90 minutes.
The prerequisites are: a second course in programming, and
recommended some knowledge of OO modeling and basic
concepts of statistics.

Materials needed are: handouts of the slides to be used in the
tutorial, and instructions on the accessing the Web page with all
the additional material on modeling and simulation with OOSimL.
Additional material (software and documentation) is located on
the following web page:
http://science.kennesaw.edu/~jgarrido/psim.html
3. LIST OF TOPICS

(Topics marked (*) will only be briefly introduced due to time constraints)

1. Real-world systems and models. (*)
2. Object-Oriented modeling. Active and passive objects. (*)
3. Discrete-event simulation and the process interaction approach. (*)
4. Random events. (*)
5. Introduction to object-oriented simulation with OOSimL.
6. General structure of an object-oriented simulation model in OOSimL.
7. Fundamental performance metrics.
8. Simple queuing models. Examples of models with simple queues.
9. Simple resource allocation. Examples of models with resources.
10. Process interaction and simple process cooperation. Examples. (*)
11. Synchronous cooperation. Examples. (*)
12. Interaction with conditional waiting. Examples. (*)
13. Process interrupts. Examples. (*)

4. REFERENCES