Proceedings of the 1983
Winter Simulation Conference
S. Roberts, J. Banks, B. Schmeiser (eds.)

429

MODEL SPECIFICATION WITH RESPECT TO ANALYSIS

C. Michael Overstreet
Computer Science Department
The University of Alabama
University, AL 35486

RESEARCH SUMMARY

Much potential exists for reducing the costs while also improving the quality of simulation models through the use of Model Development Systems (MDSs). We describe a specification language for discrete event models to be used in a MDS.

Our approach is to introduce an intermediate form between a conceptual model (the model as it exists in the mind of the modeler) and an implementation of that model in some programming language. The specification language is structured to support error analysis of both partially and completely specified models. Analysis of a model specification can also provide useful model documentation and implementation assistance.

The model specification language described here is intended to be used in conjunction with a model specification generator (a program to assist a modeler or modeling team with the construction of model specifications). Thus the language is structured to simplify model analysis rather than to be dealt with directly by a modeler.

A model is viewed as consisting of a set of objects, each object characterized by a collection of attributes. Each attribute contains information about the state of the object useful for the simulation.

A basic set of simulation primitives for specifying attribute value changes and time sequencing is defined. In order to specify the behavior of a model, each model is regarded as a collection of submodels. The actions of each submodel are then specified by the simulation primitives or by another collection of submodels, each of whose behavior must also be defined. Thus each completed model specification has a tree structure. The behavior of each interior node of the tree (each node represents one submodel) is defined by the behavior of its children (other submodels). The behavior of the terminal nodes (submodels) is specified using the simulation primitives. To allow error analysis of incomplete model specification, the "information flow" into and out of each submodel should be specified by the modeler if the behavior specification of the submodel is incomplete.

A simple example is presented using this specification language. The example is used to illustrate some types of model analyses and documentation.