AN INTRODUCTION TO THE
SIMULATION DATA LANGUAGE

Charles R. Standridge, Ph.D
A. Alan B. Pritsker, Ph.D., P.E.
Pritsker & Associates, Inc.
P.O. Box 2413
West Lafayette, Indiana 47906
317-463-5557

Abstract

In this lecture, basic concepts for using database management capabilities in simulation studies will be discussed. The implementation of these concepts in the Simulation Data Language (SDL) will be presented. SDL data organization constructs, especially those for model outputs, and SDL commands will be discussed. Simulation projects which have used SDL will be reviewed.

INTRODUCTION

As simulation has become widely accepted as an analysis technique, the complexity of the studies performed using simulation has increased. Increased complexity has made other activities in addition to model building and implementation more significant. One such activity is the management of the data involved in a large scale simulation study. Analysts often spend a great deal of time organizing input data, determining how to analyze model outputs, and studying printed results of runs. However, there has been no simulation-specific software support for data management. Thus, an analyst must develop data management software for specific projects or expand and rerun simulation programs when different output reports become necessary.

In general data processing, analysts use software tools (called database management systems) to assist with the management and organization of large amounts of data. Many such systems are widely available (2,3,14). Typically, these systems do not provide all the specific capabilities needed in simulation. A database management system for simulation must help the user organize, store and retrieve, and report input parameter values, model outputs, and analyses of model outputs. Furthermore, the system must be usable from within programs written in a simulation language. Of course, the system must still provide traditional database functions such as loading, editing, deleting and locating data that satisfies particular relationships.

We have developed SDL, the Simulation Data Language, to meet the specific data management requirements of simulation analysts. In particular, SDL provides a simulation-specific framework for organizing data and a set of commands for manipulating data in ways typically needed in simulation.

SDL DATA ORGANIZATION

SDL provides both a general data organization and a data organization for model outputs and analyses of model outputs. The general data organization is based on the relational data organization developed by Codd (1). A relation is a matrix of rows and columns. A database consists of one or more relations. The data organization for model outputs and analyses of model outputs is an extension of the relational data organization. Special structures for organizing, in the same database, outputs from different runs of the same model or from runs of different models are provided. Furthermore, special structures for organizing statistics, histograms, traces of entity behavior, and traces of system state changes are supplied.

SDL COMMANDS

SDL consists of over 120 commands, which can be organized into the following 13 categories:

1. Relation definition: Commands by which the user describes the characteristics of

Proceedings of the 1982
Winter Simulation Conference
Highland * Chao * Madrigal, Editors
82CH1844-0/82/0000-0617 $00.75 © 1982 IEEE.
the relations comprising the database
(number of columns, relation name, column names, columns containing key values);
2. Loading: Commands which transfer data
from non-SDL files to an SDL database;
3. Editing: Commands which examine data
for conformity to user specified condi-
tions;
4. File creation: Commands which transfer
data from an SDL database to a non-SDL
file;
5. Deletion: Commands which remove data
from an SDL database;
6. Statistics: Commands which compute and
store in the database commonly used sta-
tistical measures;
7. Histograms: Commands which build histo-
grams and store them in the database;
8. Report generation: Commands which gen-
erate columnar reports or display histo-
grams in either an SDL defined or a user-
specified format;
9. Plot generation: Commands which gener-
ate plots in either a user-specified or an
SDL defined format;
10. Error recovery: Commands which allow
the user to restore a correct database af-
after an error occurs during a run of an SDL
program;
11. Housekeeping: Miscellaneous commands
such as those which open and close a data-
base;
12. Storage: Commands which store one row
of data in the database; and
13. Retrieval: Commands which retrieve
one row of data from the database.

Although some commands perform general
database management functions, the impor-
tant aspect of SDL is the inclusion of com-
mands to perform functions that are uniquely
useful in simulation. These are:
1. defining relations which contain model
traces, histograms or statistics;
2. computing statistics or building histo-
grams by replication, by batches within
one replication, or by regeneration cycle,
deleting observations from an initial
transient period if desired;
3. generating reports showing statistics
concerning the same variable from models
of different system alternatives or from the
actual system;
4. generating plots displaying time series
of the same variable from models of dif-
ferent system alternatives or from the
actual system;
5. storing individual observations or sta-
tistical summaries of observations of model
variables;
6. aggregating statistics concerning a var-
iable across model replications or
batches within one replication;
7. computing the variance of sample means
of model variables using the method of re-
plications, batch means or regeneration;
8. taking a random sample from a histogram
stored in the database.

SDL also provides the interfaces for a
user to write programs for application-spe-
cific manipulations of model outputs, in-
cluding links to graphical display soft-
ware.

These commands, along with the data
organization for model outputs and ana-
lyses of model outputs, provide simulation-
specific capabilities which make SDL unique
among database management systems.

In addition to these simulation-speci-
cific capabilities, a user may make use of
the general database management capabili-
ties of SDL to enhance the modeling and
analysis process. For example, a time his-
tory of actual arrivals to a system can be
loaded into the database and edited.
This historical arrival pattern could then
be employed within the model by using SDL
retrieval commands to obtain the next ar-
rival from the database whenever it was
required.

ACCESS TO AN SDL DATABASE

Access to an SDL database is gained by
invoking SDL commands. SDL is written en-
tirely in ANSI standard FORTRAN. Thus, an
SDL command may be used by invoking the ap-
propriate FORTRAN subprogram. Typically,
this method of access is used to store and
retrieve data, within a simulation program,
for example. Alternatively, the SDL Opera-
tion Invocation Language (OIL) may be
used. The OIL is a high level programming
language designed to allow a user-friendly
interface for specifying values for the
parameters of SDL commands. Typically, OIL
programs are written to define the data or-
ganization, load data, edit data, delete
data, create sequential files, compute sta-
tistics, build histograms, generate re-
ports, and generate plots.

SDL APPLICATIONS

To date, ten SDL applications have
been performed. In these applications, SDL
has been used to:
1. perform the analysis and presentation of
model outputs;
2. allow data gathered from the system
being studied to be employed in the simu-
lation model of that system;
3. manage model input parameters;
4. decompose large-scale, complex models
into sets of smaller, simpler models; and
5. develop automatic, user-transparent data
collection in special purpose simulation
languages.
The applications include the assessment of operational policies and rates used by a capitation reimbursement system (15); the assessment of a hospital clinic and related laboratory operations (11); the projection of the availability of primary care physicians in Indiana (12); the assessment of the ability of a maintenance facility to meet scheduled maintenance requirements for a fleet of aircraft (4,5); the assessment of the ability of a job shop to meet its current production requirements (8); assessment of memory requirements for a micro-computer travelling aboard the space shuttle (14); the development of a simulation language tailored for modeling steel operations (11); the development of a transition path analysis language for economic, technological and societal problem assessment (11,17); and the development of a network simulation language for modeling river systems (6,7). In these applications SDL was helpful when:

1. comparing data generated by models of different alternatives;
2. all analyses of the output data cannot be defined before runs are made;
3. using data as input to graphics displays;
4. models are large and dividing them into components is helpful;
5. a large number of model input parameters must be managed and modified to form inputs for models of different alternatives;
6. data gathered from the system must be managed and analyzed; and/or
7. comparisons of data gathered from the system with model-generated data must be made.

**SUMMARY**

In this paper, SDL concepts for using database management capabilities in a simulation study have been overviewed. The SDL data organization, SDL commands, ways of accessing an SDL database, and applications of SDL have been surveyed.

**BIBLIOGRAPHY**


2. Date, C.J. (1975), An Introduction to Database Systems, Addison-Wesley, Reading, MA.


