Proceedings of the 2004 Winter Simulation Conference
R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

A DATA WAREHOUSE ENVIRONMENT FOR STORING AND ANALYZING SIMULATION OUTPUT DATA

Christos Vasilakis
Elia El-Darzi
Panagiotis Chountas

Harrow School of Computer Science
Northwick Park, Watford Road
University of Westminster
Harrow, HA1 3TP, U.K.

ABSTRACT

Discrete event simulation modelling has been extensively used in modelling complex systems. Although it offers great conceptual-modelling flexibility, it is both computationally expensive and data intensive. There are several examples of simulation models that generate millions of observations to achieve satisfactory point and confidence interval estimations for the model variables. In these cases, it is exceptionally cumbersome to conduct the required output and sensitivity analysis in a spreadsheet or statistical package. In this paper, we highlight the advantages of employing data warehousing techniques for storing and analyzing simulation output data. The proposed data warehouse environment is capable of providing the means for automating the necessary algorithms and procedures for estimating different parameters of the simulation. These include initial transient in steady-state simulations and point and confidence interval estimations. Previously developed models for evaluating patient flow through hospital departments are used to demonstrate the problem and the proposed solutions.

1 INTRODUCTION

The benefits of employing data warehousing techniques and On-Line Analytical Processing (OLAP) tools in preparing data for estimating the input parameters of various decision models has already been demonstrated (Koutsoukis, Mitra and Lucas 1999, Vasilakis, El-Darzi and Chountas 2004). Although streaming simulation output data to a data warehouse has been mentioned before (Banks 1997), there exists no literature on the database issues pertaining to the design of such a data warehouse. The aim of this paper is to fill this gap.

This paper is organized as follows. We begin by giving a brief introduction to recent developments in the database technology for analytical processing before giving a brief introduction to output analysis in steady-state simulations. In section 3 we examine the dimensionality of simulation output data. In section 4 we describe the theoretical models that underpin the data warehouse environment while in section 5 we demonstrate a prototype software application. Finally, a summary and conclusions can be found in section 6.

2 BACKGROUND

2.1 Data Warehousing and OLAP Systems

The database technology for supporting analytical processing as opposed to transactional processing has already shifted from the standard relational model and On-Line Transaction Processing systems (OLTP) to multidimensional models and On-Line Analytical Processing systems (OLAP) (Chaudhuri, Dayal and Ganti 2001). OLAP is a category of software technology that provides fast, consistent and interactive access to a wide variety of possible views of information. OLAP applications are generally characterized by the representation of data into multidimensional perspectives and the ability to formulate complex, ad-hoc queries that often use statistical formulae to aggregate data (Codd, Codd and Salley 1993). These multidimensional views provide quick access to historical information, usually stored in a data warehouse.

At the conceptual level of data warehouses, data are seen either as facts with their associated numerical measures or as dimensions that characterize these facts. Relational-dimensional modelling, where data are organized in a star or snowflake schema, is often preferred to entity relationship (ER) modelling as the conceptual design technique (Kimball 1998). A relational-dimensional model contains the same information as a typical ER model but the representation of the data is optimized for user understandability and analytical query performance. A star schema is a logical structure that has a factual table in the centre surrounded by dimension tables containing reference data for the facts (Connoly and Begg 2002). A snowflake schema is a variant of the star schema where the dimension tables are normalized and thus, the hierarchy of the dimensions can be depicted. The choice between the two schemata depends mainly on performance-related issues.

If the data are to be stored in a relational database (ROLAP architecture), then the above relational-dimensional model is sufficient. Once this schema has been implemented and populated with data, standard SQL can be used to extract information from it.

In multidimensional databases (MOLAP architecture) however, data are stored in n-dimensional arrays (Vassiliadis and Sellis 1999). The "data cube" is almost universally accepted as the underlying logical construct to conceptualize these multidimensional databases (Thomas and Datta 2001). It plays the same role as the "relation" in relational databases. There exists however, no universally accepted notation to describe a data cube. The model and notation proposed by Thomas and Datta (2001) for conceptualizing data cubes has been shown to have enough descriptive power without being overly complicated (Vasilakis 2003, Vasilakis, El-Darzi and Chountas 2004). It is also employed here for describing the relevant data cubes and formulating OLAP user requests.

2.2 Simulation Output Analysis

2.2.1 Initial Transient in Steady-State Simulations

According to Alexopoulos and Seila (1998), there are two types of simulations with respect to output analysis: the finite horizon, also known as terminating (Law and Kelton 2000, Banks, Carson and Nelson 2001), and the steady-state or non-terminating. In the former, the simulation starts from an empty and idle state and is run until a terminating event occurs. The purpose of a steady-state simulation on the other hand, is that of studying the long-run behaviour of the system. In such a simulation there is no natural event to signify the length of each run of the model. The performance measures of such simulations, which are characteristic of the equilibrium distribution of an output stochastic process, are called steady-state parameters (Law and Kelton 2000).

One of the main issues in steady-state simulations is the estimation of the initial transient (also referred to as initial bias or warm-up period). A simulation is said to have reached steady state when it is in a state of dynamic equilibrium in which the effects of the starting conditions have been lost (Pidd 1998).

A number of methods exists for estimating the warm-up period. Robinson (2002) arranged them in the following categories: graphical methods, heuristics approaches, statistical methods, initialization bias tests and hybrid methods. Graphic methods concern the visual inspection of output data plotted as time series. Heuristics approaches rely upon simple rules while statistical methods apply some statistical principles for estimating the warm-up period. Initialization bias tests detect possible presence of initial bias in the output data and thus they have to be combined with another method. Hybrid methods combine graphical or heuristic methods with some initialization bias tests. In addition to these, a new method that is based on the principles of statistical process control has recently been proposed (Robinson 2002). In this method, a simulation in steady-state is considered to be "in-control", whereas it is considered to be "out-of-control" during the transient phase.

The graphical procedure of Welch (1983) is considered to be the most popular method of estimating the warm-up period. The main reasons for its popularity are generality and relative ease of implementation (Alexopoulos and Seila 1998, Law and Kelton 2000). The Welch method states that for defining the initial transient period, ℓ:

- Make n replications ($n \geq 5$) of $i = 1,2,...,m$ length (where m is large)
- Average the ith observation of each replication
- Smooth out the high-frequency oscillations by defining a moving average of window w (where $w \in N+$ and $w \leq m/2$)
- Plot the moving average for several values of w and choose the smallest value of w for which the plot is reasonably smooth (by trial and error)
- Choose ℓ to be the value of i beyond which the series appears to have converged.

This method can be applied to almost any type of steady-state simulation. However, albeit easy to implement when there are few observations, problems arise when the number of required observations per replication runs to the thousands. A possible solution to this problem is described in section 4.3.

2.2.2 Point and Confidence Interval Estimators

Law and Kelton (2000) have identified six different procedures, all of which seek to eliminate or at least reduce the effects of autocorrelation: replication/deletion (also known as independent replications or IR), batch means (BM), autoregressive, spectral, regenerative, and standardized time series. The IR and BM methods are the most widely used and are further explained below.

The IR approach requires k independent runs of length n (Alexopoulos and Seila 1998). The warm-up period, ℓ, can be determined with any of the methods described above. After ℓ has been deleted from each run and letting x_{ij} be the ith observation of the jth simulation run, the following equations can be used to compute the sample means (equation 1), unbiased estimators for the steady-state mean μ (2) and sample variance (3):

$$Y_j = \frac{1}{m-l}\sum_{i=l+1}^{m} x_{ij},\ j = 1,2,\ldots,n \tag{1}$$

$$\bar{Y}_k = \frac{1}{k}\sum_{i=1}^{k} Y_i \tag{2}$$

$$S_k^2 = \frac{1}{k-1}\sum_{i=1}^{k}\left(Y_i - \bar{Y}_k\right)^2 . \tag{3}$$

Additionally, given sufficiently large n and k an approximate 1-α confidence interval for μ can be computed by:

$$\bar{Y}_k \pm t_{k-1,1-\alpha/2}\frac{S_k(Y)}{\sqrt{k}} \tag{4}$$

The IR approach is believed to give good statistical performance, is easy to implement, and can be used to estimate different parameters of the model (Law and Kelton 2000). However, since data must be deleted from each replication this approach is believed to be uneconomical (Banks, Carson and Nelson 2001).

The BM method overcomes the problem of having to go through the warm-up period many times by having only one long replication of the model. Further, it is believed that the BM estimators for μ converge faster than the respective IR (Alexopoulos and Goldsman 2003). The output of this single long replication of n observations is then divided into batches, which can be treated as independent (Banks, Carson and Nelson 2001). The batches are formed after the initial transient, ℓ, has been deleted. Figure 1 illustrates this concept for k batches of size $m = (n - \ell) / k$, where x_i is the ith observation of the single long replication.

$$\underbrace{X_1,\ldots,X_\ell}_{deleted},\underbrace{X_{\ell+1},\ldots,X_{\ell+m}}_{\bar{Y}_1},\ldots,\underbrace{X_{\ell+(k-1)+1},\ldots,X_{\ell+km}}_{\bar{Y}_k}$$

Figure 1: The Batch Means Method, Adapted from Banks, Carson and Nelson (2001)

The mean of the batches is computed by equation (5) and the variance of the sample mean by (6) (Banks, Carson and Nelson 2001). As before, (4) can be used to compute the 100(1-α)% confidence interval for μ, where $0 < \alpha < 1$.

$$\bar{Y}_j = \frac{1}{m}\sum_{i=(j-1)m+1}^{jm} X_{i+\ell},\ j=1,2,\ldots k \tag{5}$$

$$S_k^2 = \sum_{j=1}^{k}\frac{\left(\bar{Y}_j - \bar{Y}\right)^2}{k-1} \tag{6}$$

where

$$\bar{Y} = \sum_{j=1}^{k}\bar{Y}_j .$$

3 THE DIMENSIONALITY OF SIMULATION OUTPUT DATA

In examining the dimensionality of simulation output data, the emerging critical dimension is that of "Model", which has also been suggested to be considered as a dimension (Koutsoukis, Mitra and Lucas 1999). Generally, for any decision modelling situation, such a dimension can be used to conceptualize the concept of having different model versions or scenarios. These model versions can be conceived as part of either "what-if" or sensitivity types of analyses. It can be assumed that there is a two-level hierarchy associated with such a dimension, the root level being the "Model" and the second level being the "Scenario". The former, can be thought of as the generic level of analysis, while the second can be thought of as the version-control level (e.g., different versions of the same model with different parameter values).

In simulation modelling however, two more levels must be introduced. Considering that, in the IR method for estimating the steady-state parameters of a simulation model each version of a model (scenario) requires n replications of length m to be run, the levels "Replication" and "Observation" need to be introduced to the "Model" dimension. Figure 2 illustrates graphically the levels of the hierarchy and instances of dimension values. The tree structure of the dimension translates to the following: a model comprises several scenarios, each scenario requires n replication to be executed, while each replication is of length m.

Having described the dimension "Model", a simple example can used to illustrate the necessity for employing data warehousing techniques in analyzing simulation output data. Assume a simulation model is executed by the IR method and requires 20 different scenarios, 20 replications per scenario, 5000 steady-state days per replication, and 3 output parameters. If we also assume that we require one observation per simulated day, then the total number of observations is 2 million, while the total number of data items are 6 million. Clearly, this figure suggests the use of database technologies since spreadsheets and most statistical packages cannot cope with this amount of data.

A second dimension in the schema is that of simulated time or "Clock". This dimension can be extremely useful in, for example, the BM method by facilitating the estima-

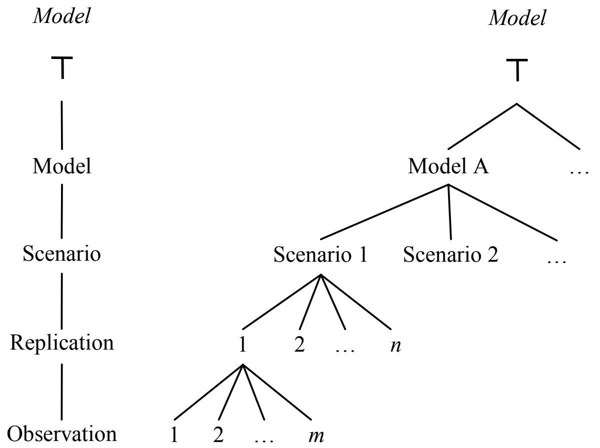

Figure 2: Schema (Left) and Instances (Right) of the "Model" Dimension

tion of output parameters. Moreover, it can be useful in simulations where some trigger exists that temporarily changes the conditions of execution and thus, the estimation of the parameters for different time periods is needed. In both cases the hierarchy of the dimension contains three levels: the root, the "period" i.e. the grouping level, and the actual time unit ("clock") of the simulation (Figure 3). In the BM method for instance, the period level can be associated with information regarding the batches.

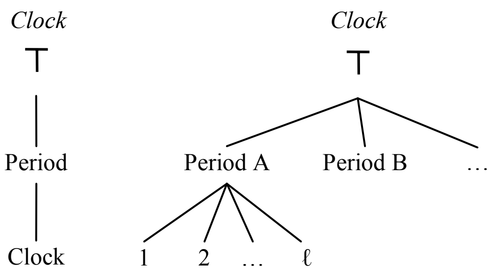

Figure 3: Schema (Left) and Instances (Right) of the "Clock" Dimension

4 DATABASE MODELS

4.1 Analytical Requirements

In this section, we propose a relational and a multidimensional data model for storing simulation output data as part of an innovative approach in which, the main objective is to facilitate the necessary procedures for estimating the output parameters of the simulation. More specifically, we want to facilitate the implementation of the Welch graphical method for estimating the initial transient in steady state simulations (Welch 1983).

Additionally, we seek to facilitate the estimation of the sample means (equation 1), unbiased estimators for μ (2) and sample variance (3) for the IR method, and the BM (5) and variance of the sample mean (6) for the BM method. Finally, the estimation of the 1 - α confidence intervals for μ (4), which is common in both methods.

We use a previously described discrete simulation model of patient flow as a running example (El-Darzi et al. 2000, Vasilakis and El-Darzi 2001). This simulation model exhibits particularly long initial transient periods and long replications or batches are required for an accurate estimate of its steady-state parameters. This phenomenon is due to the considerable longer length of stay of the patients that finally reach the long-stay compartment (Figure 4). The two queues between the compartments measure bed blockage in the system while the external queue measures the time arriving patients may have to wait before being admitted to hospital. For simplicity, in this paper we are only interested in three system-state parameters, the number of patients in each of three compartments (short, medium and long-stay). Based on this simplification, we assume that stable-state has been reached when the time-series of the total number of patients in the system has converged.

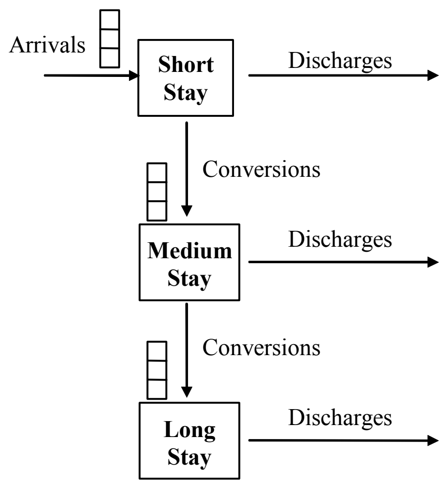

Figure 4: Network Representation of the Simulation Model of Patient Flow

4.2 Relational-Dimensional Model for Simulation Output Data

A snowflake schema for storing output simulation data is proposed in this section, while the simulation model of patient flow briefly described in section 1 is used to demonstrate the usability of the environment. The snowflake schema is preferred to the star schema in this instance, as dimension "Model" conveys metamodel information about the simulation models, scenarios and replications (see Figure 5). As mentioned in section 2.1, ultimately the choice between star and snowflake schema is down to performance-related issues that are beyond the scope of this paper.

Attributes *serial_no* and *random_seed* of the Replication entity store, respectively, the serial number of the replication and the random seed that is used to initialize the

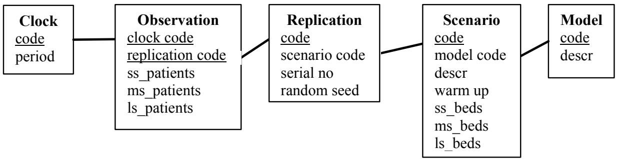

Figure 5: Relational/Dimensional Schema (Snowflake) for Simulation Output Data

random number generator. Attribute *warm_up* stores the warm up period once it has been estimated, while attributes *ss_beds*, *ms_beds*, *ls_beds* store the number of beds in the capacitated model. All these attributes are stored at the Scenario level. Attribute *descr* in entities Scenario and Model stores a textual description at the model and scenario levels.

The fact entity, Observation, has a composite primary key comprising of attributes *clock_code* for linking with the Clock dimension and *replication_code* for linking with the Replication entity, the first level of the Model dimension. For simplicity, we assume the values of the attribute *replication_code* to be unique in the database. Attributes *ss_patients*, *ms_patients*, and *ls_patients* contain the values of the three simulation parameters (short, medium and long-stay patients in each compartment) as sampled by the simulation engine on each simulated day.

Having described the relational model we now demonstrate how standard SQL can be used to facilitate the estimation of the required parameters. The following view is used for calculating the steady-state sample means of parameter *ss_patients* for each replication in the IR method (1):

```
CREATE VIEW      sample_means
AS      SELECT   r.scenario_code,
                 o.replication_code,
                 AVG(o.ss_patients) AS mean
        FROM     observation o,
                 replication r, scenario s
        WHERE    o.replication_code = r.code
        AND      r.scenario_code    = s.code
        AND      o.clock            > s.warm_up
        GROUP BY r.scenario_code,
                 o.replication_code;
```

If the batch/means method has been used and assuming that column “Period” of table “Clock” is populated according to the length of the batches, the following view can be used to calculate the sample means of the batches:

```
CREATE VIEW      sample_means
AS      SELECT   r.scenario_code, c.period,
                 AVG(o.ss_patients)
                             AS sample_means
        FROM     observation o, scenario s,
                 replication r, clock c
        WHERE    o.replication_code = r.code
        AND      r.scenario_code    = s.code
        AND      o.clock_code       = c.code
        AND      o.clock            > s.warm_up
        GROUP BY r.scenario_code, c.period;
```

The following SQL statement can now be used to calculate the unbiased estimator for μ and sample variance for each scenario in the database, either for the IR or the BM approach:

```
SELECT     scenario_code, AVG(mean),
           VAR(mean)
FROM       sample_means
GROUP BY   scenario_code;
```

4.3 Data Cube Model for Simulation Output Data

In this section we define a data cube model that can be utilized in cases where a multi-dimensional database is preferred to a relational one for the data warehouse. With reference to the dimensions and hierarchical structures described in section 3, the analytical requirements set out in section 4.1, and according to the notation in (Thomas and Datta 2001), a data cube called OUTPUT can be defined as follows:

C={observation, model, clock};
d(model)=1, d(clock)=1, and d(observation)=0;
D={clock, replication, par, period, scenario, model},
M={ss_patients, ms_patients, ls_patients);
f(model)={replication, scenario, model},
f(clock)={clock, period},
f(observation)={par};
O_{model}={⟨replication_code, scenario_code⟩,
⟨scenario_code, model_code⟩},
O_{clock}={⟨clock, period⟩} and,
$O_{observation}$={}

where *par* is the output parameter of the simulation.

This data cube model can be utilized in applying the Welch graphical method for estimating the initial transient period (see section 2.2.1). The following OLAP algebra statement averages the *i*th observation of each replication and constructs a moving average of window w at the scenario level:

$$\Gamma_{[\mathrm{MA(w)},\{\mathrm{scenario}\},\Gamma_{[\mathrm{average},\{\mathrm{replication}\},\mathrm{par}]}} \ldots(\mathrm{OUTPUT}) = C_R$$

The innermost aggregate function (average) is performed at the “Replication” level of the “Model” dimension (for averaging the *i*th observation of each replication) while the moving average function is performed at the “Scenario” level. Naturally, different windows can be easily implemented to facilitate the trial-and-error approach of the Welch method. Once the point of convergence, ℓ, has been established the unbiased estimator for μ (equation 2) can be calculated by the following statement:

$$\Gamma_{[\text{average}, \{\text{scenario}\}}, \Gamma_{[\text{average}, \{\text{replication}\}, \text{par}]}]$$
$$\ldots \Sigma_{\text{clock} > \ell}(\text{OUTPUT}) = C_R$$

The innermost aggregate function that is performed at the "Replication" level essentially gives the sample means for each replication (equation 1), while the outmost function gives the unbiased estimator for μ. In similar fashion, assuming that "var" is the standard SQL function for calculating the variance, the following statement provides the unbiased estimator for the sample variance:

$$\Gamma_{[\text{var}, \{\text{scenario}\}}, \Gamma_{[\text{average}, \{\text{replication}\}, \text{par}]}]$$
$$\ldots \Sigma_{\text{clock} > \ell}(\text{OUTPUT}) = C_R$$

4.4 Implementation Issues

The implementation of the dimensional models described in this paper can take place in any commercial relational database management system (RDBMS) such as Oracle or SQL Server 2000, or an OLAP server that supports the ROLAP architecture. In either case, standard SQL can be used to perform the analytical queries. The relational tables can be easily populated with simulation output data by uploading the standard output files generated by the simulation engine or software package (e.g. in the case of Arena the `.dat` files, or in the case of Micro Saint the `.res` files).

On the other hand, the data cubes can be implemented using an OLAP server that supports the MOLAP architecture. The data cube model reported here was implemented on Microsoft's Analysis Services while the OLAP queries were written in Multidimensional Expressions (MDX), a language specifically designed for OLAP analysis (Spofford 2001). The front-end of the OLAP prototype systems has been implemented with the Pivot Table Services of Excel. A typical MDX expression for a moving average (w = 40) on variable "ls_patients" is as follows:

```
avg({[Time].CurrentMember.Lag(20):
    [Time].CurrentMember.lag(-20)},
        [Measures].[LS Patients])
```

A screenshot from the prototype application is illustrated in Figure 6. On the left-hand side are the dimensions and measures available to the user, while in the main pivot-chart area two parameters are displayed in a time-series format, the total number of patients in the system and a moving average (w = 40) of the same parameter (bold line). The "Model" dimension is used as a filter dimension, while the "Clock" dimension is used on the x-axis of the plot.

5 CONCLUSIONS

In this paper we propose a database environment for analyzing simulation output data. The proposed database environment is capable of providing the means for automating the necessary algorithms and procedures for estimating the parameters of the simulation. It can be readily employed for estimating the initial transient in steady-state simulations. Several moving averages with different windows can

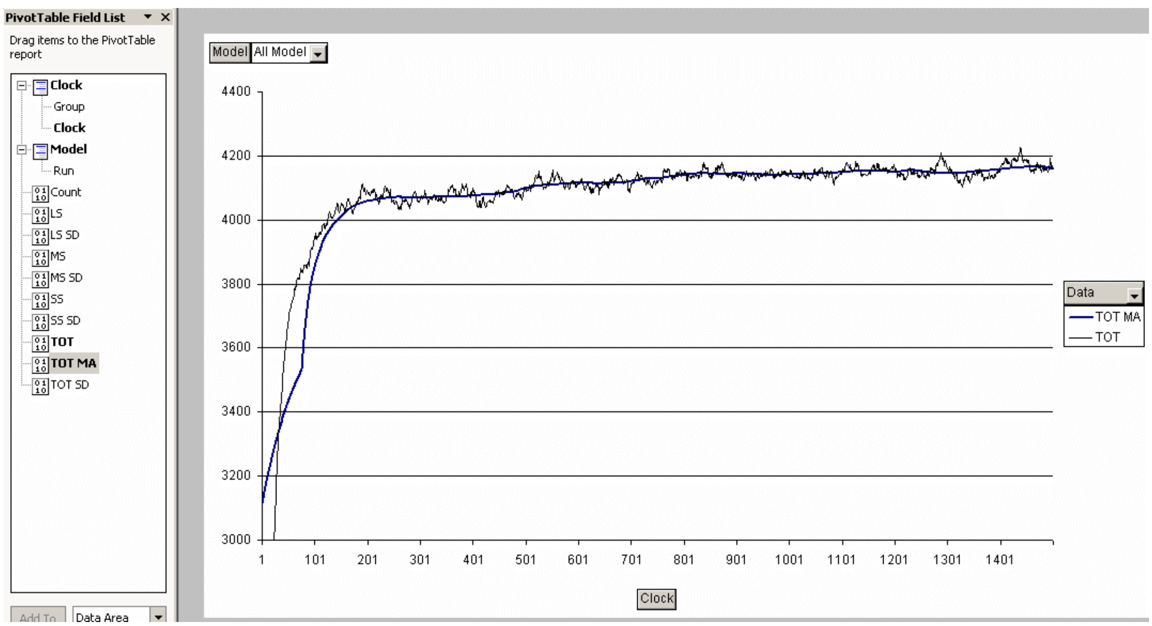

Figure 6: Screenshot of the Prototype Application

be easily plotted once they have been calculated. Parameters such as the length of the initial transient, and the length and number of batches can be easily changed. Relational or multidimensional queries can also be used to accommodate the estimation of more complex performance measures. They may include measures such as average utilization in the system, average spare capacity, and parameters relating to the queuing measures.

Such an environment can substantially reduce the computational complexities in analyzing and interpreting simulation output data. The capability of storing information about the simulation models (random seed, initial values of system-state parameters) is another important feature of the proposed environment. By designing and developing a novel data warehouse and OLAP environment for analyzing simulation output data we took a step toward rendering simulation engines "black boxes" for the end-users.

Further work is needed, particularly in testing how this environment can be extended to other methods for detecting the end of the warm-up period and estimating steady-state parameters. In addition, the seamless integration of this data warehouse environment with existing simulation packages could be of particular interest in future research and development endeavors.

REFERENCES

Alexopoulos, C. and D. Goldsman. 2003. To Batch or Not to Batch. In *Proceedings of the 2003 Winter Simulation Conference*, eds. S. Chick, P. J. Sánchez, D. Ferrin and D. J. Morrice, 481-489. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Alexopoulos, C. and A. F. Seila. 1998. Output Data Analysis. In *Handbook of simulation: principles, methodology, advances, applications and practice*, ed J. Banks, 225-272. New York: Wiley.

Banks, J. 1997. The Future of Simulation Software: A Panel Discussion. In *Proceedings of the 1997 Winter Simulation Conference*, eds. S. Andradottir, K. J. Healy, D. H. Withers and B. L. Nelson, 1155-1162. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Banks, J., J. S. I. Carson and B. L. Nelson. 2001. *Discrete-event system simulation*. New Jersey: Prentice-Hall.

Chaudhuri, S., U. Dayal and V. Ganti. 2001. Database Technology for Decision Support Systems. *Computer* 34(12): 48-55.

Codd, E. F., S. B. Codd and C. T. Salley. 1993. *Providing OLAP to User-Analysts: An IT Mandate*. Sunnyvale, California: E.F. Codd & Associates.

Connoly, T. and C. Begg. 2002. *Database Systems, A practical approach to Design, Implementation , and Management*. Essex: Addison-Wesley.

El-Darzi, E., C. Vasilakis, T. Chaussalet and P. H. Millard. 2000. A Simulation model to evaluate the interaction between acute, rehabilitation, long-stay care and the community. In *Decision Making: Recent Developments and Worldwide Applications*, eds. S. H. Zanakis, G. Doukidis and C. Zopounidis, 475-485. Dordrecht: Kluwer Academic Publishers.

Kimball, R. 1998. *The data warehouse lifecycle toolkit: expert methods for designing, developing, and deploying data warehouses*. New York: Wiley.

Koutsoukis, N.-S., G. Mitra and C. Lucas. 1999. Adapting on-line analytical processing for decision modelling: the interaction of information and decision technologies. *Decision Support Systems* 26(1): 1-30.

Law, M. A. and W. D. Kelton. 2000. *Simulation Modelling and Analysis*. Singapore: McGraw-Hill.

Pidd, M. 1998. *Computer simulation in management science*. Chichester: Wiley.

Robinson, S. 2002. A Statistical Process Control Appoach for Estimating the Warm-Up Period. In *Proceedings of the 2002 Winter Simulation Conference*, eds. E. Yücesan, C.-H. Chen, J. L. Snowdon and J. M. Charnes, 439-446. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Spofford, G. 2001. *MDX Solutions with Microsoft SQL Server Analysis Services*. USA: John Wiley & Sons.

Thomas, H. and A. Datta. 2001. A conceptual model and algebra for on-line analytical processing in decision support databases. *Information Systems Research* 12(1): 83-102.

Vasilakis, C. 2003. Simulating the flow of patients: an OLAP-enabled decision support framework. Ph.D. thesis, Department of Computer Science, University of Westminster, London.

Vasilakis, C. and E. El-Darzi. 2001. A simulation study of the winter bed crisis. *Health Care Management Science* 4(1): 31-36.

Vasilakis, C., E. El-Darzi and P. Chountas. 2004. An OLAP-enabled environment for modelling patient flow. *Journal of Computational Methods in Sciences and Engineering* forthcoming.

Vassiliadis, P. and T. Sellis. 1999. A survey of logical models for OLAP databases. *ACM SIGMOD Record* 28(4): 64-69.

Welch, P. 1983. The statistical analysis of simulation results. In *Computer Performance Modeling Handbook*, ed S. Lavenberg, 268-328. New York: Academic Press.

AUTHOR BIOGRAPHIES

CHRISTOS VASILAKIS is a lecturer in the Harrow School of Computer Science, University of Westminster, London, UK. He obtained his M.Sc. with Distinction in Decision Sciences and Ph.D. in Computer Science from the same university. His research interests include the interaction between data modeling (mainly data warehousing and OLAP) and decision modeling techniques (mainly discrete

event simulation), and patient flow simulation models. He is a member of INFORMS and The Operational Research Society in the UK. He has recently been awarded a Michael Smith Foundation for Health Research fellowship to develop a computer simulation model to evaluate organizational changes related to access to health care. His e-mail address is <C.M.Vasilakis@wmin.ac.uk> and his web address is <www2.wmin.ac.uk/coiec>.

ELIA EL-DARZI is a senior academic in the Harrow School of Computer Science, University of Westminster, London, UK. He obtained his Ph.D. in Operational Research from Brunel University, UK. His research interests include computational optimization and the application of management science techniques to patient flow modelling. He is a member of INFORMS and The Operational Research Society in the UK. His e-mail address is <eldarze@wmin.ac.uk>.

PANAGIOTIS CHOUNTAS is a lecturer in the School of Computer Science, University of Westminster. He graduated with a B.Sc. in Computing & Electronics and was awarded a Ph.D. in the area of temporal multi-source database systems from UMIST & Victoria University of Manchester. He has co-authored over 40 papers in the area of flexible systems. He has recently served on the programme committee of the Eighteenth International Symposium on Computer and Information Sciences (ISCIS). He is also a regular reviewer for the "Information Systems" and "Requirements Engineering" journals. His e-mail address is <chountp@wmin.ac.uk>.