Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

INITIALIZATION OF ONLINE SIMULATION MODELS

André Hanisch
Juri Tolujew

Fraunhofer Institute for Factory
Operation and Automation IFF
Sandtorstrasse 22
39106 Magdeburg, GERMANY

Thomas Schulze

School of Computer Science
University of Magdeburg
Universitätsplatz 2
39106 Magdeburg, GERMANY

ABSTRACT

Online simulation is a relatively new control strategy for short-term decision-making for the control and management of processes in existing systems. In contrast to traditional "non-terminating" simulation, online simulation cannot use a transient phase to tune the models because the simulation models need to run very quickly and also need to deliver results right from the start. In this context, the initialization of such online simulation models represents a special problem. It requires mapping between the system-describing variables in the model and the available data in the real system. This paper examines two different methods of initialization. Special emphasis is placed on explaining the approach of parent model synchronization. Both initialization approaches are transferred to the context of analyzing and forecasting pedestrian flows in a public building. A first prototypical implementation in SLX is also briefly presented.

1 MOTIVATION

Decision support with a long time horizon is one of the classic applications of simulation models. Different parameters of planned systems have to be evaluated based on the results of simulations and planners endeavor to optimize these parameters. In this standard application the simulation models are offline, i.e. the models are not directly coupled with the real system. Often, the simulation models developed are not used after the decision-making process.

By contrast, short time decisions have to be made constantly when controlling and managing processes of existing systems. Nowadays, the complexity of such systems is increasing as the need to efficiently control and manage them increases. There are two main classes of control strategies: Simulation-based strategies and strategies based on heuristics rules and mathematical equations. Simulation based on the current state of the real system can provide decision makers higher quality support.

Such simulation-based online control systems require:

- A validated simulation model of the real system in which the level of detail of the simulation model must be equivalent to structures in the real system. This particularly applies to control strategies used or the incorporation of human decision making.
- An online connection of the simulation model with the real system, i.e. the inputs from the environment are identical for the real system and the model.
- The simulation including multiple different runs must be executed in a short timeframe because the results of the simulation(s) have to be available before a certain deadline. The simulation engine has to be fast enough to deliver the results in a period of time that allows using the results in the subsequent decision process.

A special problem in this context is the initialization of online simulation models. It requires mapping between system-describing variables in the model and the available data in the real system. Different problems exist where, for example, the real world measured data is incorrect or measuring the value for the model variable in the real system is problematic. Errors resulting during the initialization process have to be minimized. Normally, the consequences of such deficiencies diminish as the duration of the simulation increases. However, in contrast to traditional "non-terminating" simulation, online simulation cannot use a transient phase to tune the model because the simulation models need to deliver results right from the start.

This paper describes methodologies for initializing online simulation models in which the measured online data is either incorrect or unavailable. Section 2 of this paper examines related work on online simulation and section 3 explains different initialization methods. Section 4 describes a case study in which two of the methods described were applied. Section 5 highlights some aspects of implementation in the simulation language SLX. A summary and outlook in section 6 finish the paper.

2 RELATED STUDIES ON ONLINE-SIMULATION

Online simulation designates a category of simulation applications in which the simulation model is connected online with the reality to be simulated and the results of the simulation are available before a certain time limit expires. (Davis 1998).

The uncertainty of the simulated results correlates with the length of the simulation. The greater the difference between the forecast horizon and the current time, the more flawed the results are.

A typical area of application for online simulation (also called real-time simulation) is proactive decision support for scheduling problems in manufacturing systems (Katz and Mannivannan 1993; Harmonsky 1995; Sivakumar 1999; Gupta, Sivakumar, and Sarawgi 2002; Chong, Sivakumar, and Gay 2002; Glinsky and Wainer 2002). This scheduling practice is also known as real-time scheduling. Additional areas of application are the simulation of street traffic (Mazur, Chrobok, and Hafstein 2004) and the simulation of pedestrian flows in public buildings (Hanisch et al. 2003).

One of the fundamental tasks in online simulation is the online supply of the simulation model with the data from the real system. At the actual start of the simulation, the simulation model must reproduce the state of the real system. In a simple case, the states of all objects in the simulation model can be matched with the data from reality. The condition of data availability and correctness is not given for all applications, yet the methods for initializing simulation models described in the literature require these conditions. Data warehouses and factory databases make data available for solving scheduling problems in manufacturing systems. Fowler and Rose 2004 point out that a chief problem is the availability of on-time and correct data from the factory.

There are other applications such as traffic and pedestrian flow simulations in which collecting the required data is not possible or the collected data is incorrect.

Successful application of online simulation requires possibilities to initialize the models with sufficient precision. The longer the forecast time is, the less influence insufficient initialization has. Classic non-terminating simulation incorporates this fact in the appropriate procedures. By contrast, online simulation has a brief forecast period, i.e. insufficient model initialization inevitably leads to an erroneous forecast. Users must accept that the results are erroneous. Suitable methods can be used to reduce the magnitude of the error.

The following presents the problems of initialization in more detail and proposes different methods for initialization.

3 INITIALIZATION METHODS FOR ONLINE MODELS

In the classical simulation approach, the simulation model is initialized with “empty” and “idle” and started. Depending upon the kind of simulating system, a transient phase may be used to adjust the statistics. Consider a classical example, the simulation of counter operations in a bank. If the simulation time is designated with t, the model will be initialized with “empty” and “idle” at the time t_0. Customers stream into the bank, are served at the counters and leave the bank afterward. At a certain point in time t_E the doors of the bank are closed and the simulation terminates once the last customer has left the bank.

The online simulation starts at the time t_S, the following applying: $t_0 < t_S < t_E$.

In online simulation, the simulation model certainly can not be initialized and started as “empty” and “idle” since customers will already be in the bank. The system variables of the model have to be assigned initial values. These values must be taken over from the real system. The real data and the system variables have to be mapped so that the model can be started as a reproduction of the real system with its current state.

The real data of the system can be accessed in two ways: Direct measurement by means of sensors or access of information systems. The utilizable data is assigned two characteristics: Availability and quality.

The characteristic of availability specifies whether the data from the material system can be determined or measured at all. The following values are distinguished here: Complete and incomplete availability. When availability is complete, all necessary data is present and can be computed from existing data on the basis of context-dependent rules. By contrast, when availability is incomplete, not all necessary data can be provided, acquired or computed.

The second characteristic of quality describes the correctness of the data. A difference is made between the attributes of measuring errors and up-to-dateness. Measuring and acquisition errors are unavoidable concomitants of data acquisition. The magnitude of the error depends on the measuring and collection devices employed and the applications. For example, the results from recording persons waiting at a ticket counter contain more errors than data on machine parts inventories derived from an information system. The other attribute of up-to-dateness specifies the value of the difference between the time of model initialization and the time of measurement. This plays an important role in the provision of data in information systems. When the update frequency of this data is low, it does not reflect the current state of the real system. (Fowler and Rose 2004)

The ideal case is a combination of complete data availability and high data correctness. Established approaches to online model initialization presume the ideal case. Combinations of incomplete availability of poor quality data are more typical. This paper abstracts the lack of quality of the data, i.e. it is assumed that the data is accurate and time-equivalent at the time of retrieval. Other studies will have to examine the incorporation of the characteristic of data quality and the determination of its effects on the simulation results.

Two different methods can be applied to initialization:

- *Synchronization of a parent model* with the real system and creation of an initialized child model, and
- *Model generation* including initial settings based on existing data.

3.1 Synchronization of a Parent Model

A so-called parent simulation model is synchronized with available data. The parent model reflects the real system. One or more child models can be created at dedicated time points. These child models inherit all system variables including their values from the parent model. The simulation-based forecast is executed on one or more of these child models. Each child model has to run as fast as possible meaning it has to operate with a time advance mechanism different than its parent's.

Use of this synchronization method of a parent model requires a special simulation software functionality: Creation of a child model. In general, not all simulation software is capable of creating a copy of the running model and executing this copy with a different time advance mechanism as an independent process on the same or on another computer. Two possibilities for creating child models exist. Either the child models run as independent processes of the operating system or the child models as part of the parent are only emulated inside the parent model and do not start a new operating system process.

Figure 1 shows the first approach of using different operating system processes. This approach has the advantage that different forecasts can be executed simultaneously. The simulation system eM-plant supports such an approach.

The other possibility is to emulate the child model inside the parent model. This approach requires the simulation software be able to roll back the simulation time and switch between different time-advance mechanisms. The same capabilities are required to execute so-called nested simulations. The simulation system SLX and the old simulation language SIMULA have features that can emulate such a child model. The disadvantage of this approach is that no more than one child model can run at one time. Each has to be executed sequentially. The SLX features utilized are described in section 5 of this paper.

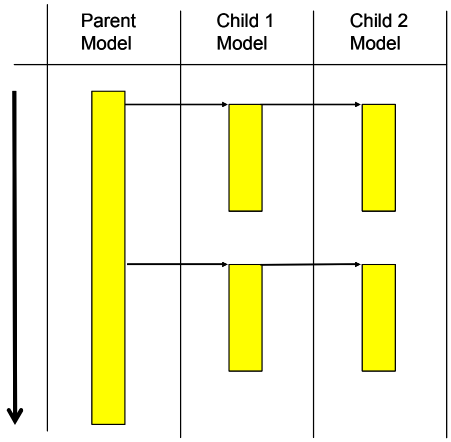

Figure 1: Child Model Execution as Different Parallel Processes of the Operating System

So far the possibilities for synchronizing the parent model with the real system have not been described. Basically, permanent and requested synchronization are distinguished. When synchronization is permanent, the parent model is continuously updated with received new values for model variables representing the state of the real world-real system variables. Therefore, a time advance mechanism proportional to real-time has to be implemented for the parent model. For a simulation-based forecast, a child model of this parent model has to be created on demand. Figure 2 shows the relations between parent model and generated child.

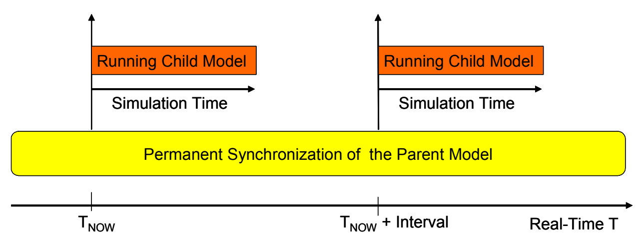

Figure 2: Child Model Creation and Execution with Permanent Synchronization of a Parent Model

Other applications also exist in which the underlying data sources are not permanently updated. The information system is updated only every five minutes, for example. In such a case, the real world data can no longer be continuously synchronized with the parent model and thus has to be synchronized upon request. This is requested synchronization. The parent simulation model "sleeps" until it is time to update the data source occurs or a request arrives. Thereupon, the parent model is updated, initiates the creation of one or more child models and starts each of the child models. Afterward, the parent model returns to its dormant state until a new update or request arrives. Figure 3 shows the basic principle with one child model.

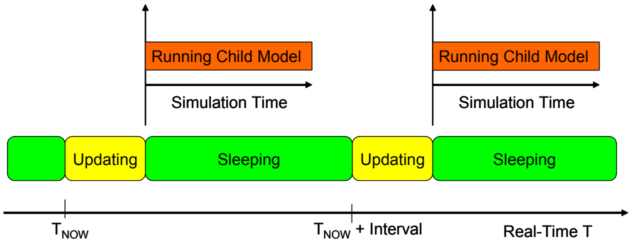

Figure 3: Child Model Creation and Execution with Requested Synchronization of a Parent Model

If the real time at which the measured data from interval k is available and at which the forecast should be started is designated with T_{NOW} and the current time of the simulation model is designated with t_i, then the following applies: $T_{NOW} > t_i$.

The parent model simulation starts at real time T_{NOW} with a time t_i and runs until the simulation time t has reached the value of T_{NOW}. Including the measured values for the inputs from the interval k, a new model state is computed in the mode "as fast as possible". Incorporating a comparison with the measured or computed values of the real system from the interval k, the model variables are then modified based on the model. No generally accepted correction instruction can be specified to do this. This correction has to be made allowing for the conditions specific to the model.

Figure 4 shows the correlation between real-time T and simulation time t in the parent model and in the child model at each synchronization request.

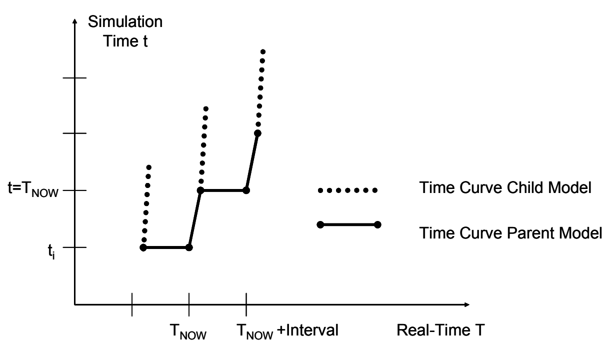

Figure 4: Correlation between Real-time T and Parent Model Simulation Time t with One Child Model

Mannivan and Banks (Mannivan and Banks 1991) also employ this synchronization approach without child model creation for exception detection.

3.2 Model Generation

Model generation is the other approach for initializing online simulation models. This approach is very similar to the classical simulation application: The online simulation model is created, initialized and started. This approach can be implemented very easily. Figure 5 shows the principle behind this approach.

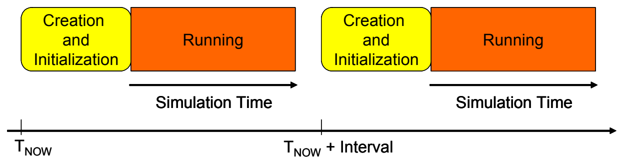

Figure 5: Model Generation including Initial Settings based on Existing Data

3.3 Comparison of Both Methods Allowing for the Incompleteness of the Data

The advantage of the model generation approach is clearly its simple implementation. Most simulation packages available can create simulation models based on formatted files. The critical disadvantage of this approach is that the data available must be complete. Missing data must be computed or assumed. The error in these computations or assumptions correspondingly affects the validity of the forecast.

By contrast, the reaction to the incompleteness of the data can be better when the synchronization approach is used for certain applications. A prerequisite is that the simulation model is completely synchronized with reality at a real time T_0. Thus, for example, it could be assumed that, if all entrances to a subway station are closed, no persons will be in the station. If a simulation is supposed to be started at the time $T_i > T_0$ then the model will be initialized based on the state at the time T_{i-1} and the available data on the real system. To estimate the missing data, two sources can be reverted to, the last state of the model and the real data. In view of the integration of the "simulated" knowledge about the system at the time T_{i-1} better quality is achieved when the initialization is at the time T_i. This knowledge is dispensed with in model generation approach.

Both approaches are presented in the following case study.

4 CASE STUDY

University of Magdeburg and the Fraunhofer Institute IFF Magdeburg jointly developed a case study for online simulation. The real system can be compared with a train station concourse entered by persons some of whom then make use of a particular service and then leave the concourse again. A portion of the persons does not make use of any service. Sensors are installed at the system's points of entry and exit. Based on direction, these counting sensors detect the number of persons who have gone through in one interval (5 minutes). Persons who make use of the services dwell in the service area in the concourse. The simulation model is used to predict the density of persons in the service area. Anytime the limit value of person density is exceeded, this should be indicated.

Figure 6 shows the simplified structure of the real system.

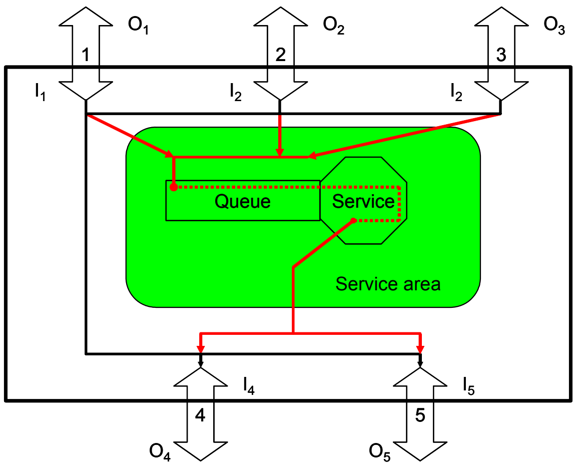

Figure 6: Simplified Real System Structure

4.1 Formal Description

The following has been defined:

- n the total number of gates, and i the index for each gate
- m the total number of time intervals and j the index for each interval
- $I_{i,j}$ the number of persons entering through gate i in interval j,
- $O_{i,j}$ the number of persons exiting through gate i in interval j
- C_j the number of persons in the system in interval j (contents)
- CP_j the total number of persons on the paths in interval j
- CS_j the total number of persons on the service area in interval j,
- CW_j the total number of persons remaining in the service area in interval j, and
- CA_j the total number of persons interacting with the service facilities in interval j.

The goal of the simulation is to forecast the total numbers of persons in the system and the number of persons waiting on the basis of the state of the real system at the real time T and further incoming persons to be anticipated.

The values for C_j und CS_j for the intervals j have to be predicted, the time T_j for these intervals being greater than the real time T_{NOW}.

The dependencies are:

$$C_j = CP_j + CS_j$$
$$CS_j = CW_j + CA_j.$$

The simulation model itself is based on a simple structure. Every person is simulated as an entity and every person entering the system is assigned the path system. The appropriate path through the system is selected on the basis of distribution functions derived from historical data.

4.2 Offline Input Data

Offline data designates data and information that do not describe the current state of the system but are necessary for the simulation. The following different kinds of offline data were employed for the case study:

- Arrival rate at the points of entry
- Departure rate at the points of departure
- Service rate on the service area
- Speed of the persons
- Distribution of the persons at the different destinations
- Path network with path lengths

Arrival rate at the Points of Entry. Counting sensors measured entries of individual persons at the points of entry. This historical data was aggregated to an arrival rate of persons per 5 minutes. The arrival rates at the points of entry exhibit typical non-stationary behavior. Figure 7 shows the number of persons measured per 5 minutes who entered the concourse through a point of entry over a 24 hour period.

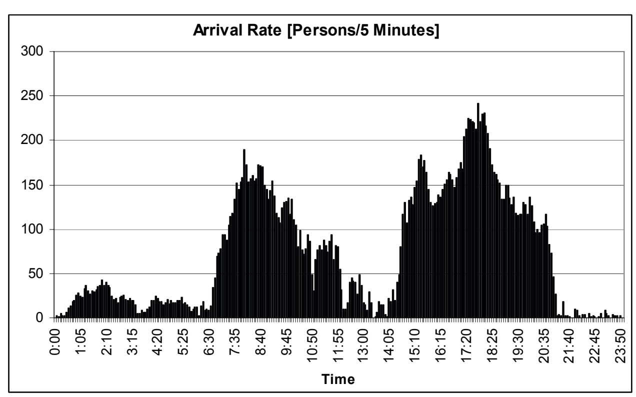

Figure 7: Arrival Rate at One Gate

This non-stationary behavior was approximated by piecewise-constant arrival rates.

Departure Rate at the Points of Departure. The departure of individual persons from the concourse was likewise measured with counting sensors installed above the points of departure. The number of persons in the system can be calculated from the total sum of persons arrived minus the total sum of persons that left the system. Figure 8 shows the number of persons in the system over a period of 24 hours. This is typical curve for the real system.

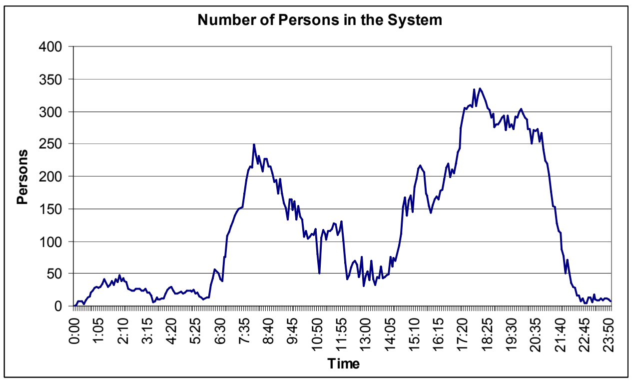

Figure 8: Number of Persons in the System (Calculated).

Service rate in the service area

The service rate (*SR*), the number of persons that can be served in one time unit, cannot be measured directly. Therefore, a service level (*SL*) is estimated based on historical data. This service level describes the probability that a person will make use of the service. The quantities of persons departing and an assumption about the service level of the system can be used to estimate the service rate in the service area. This time-dependent service rate is used for the simulation.

If the following are designated thusly:

- SR_j the service rate at interval j,
- $O_{\bullet j}$ the number of persons that have exited the concourse through any door during interval j, and
- SL the service level

then the service rate can be calculated for interval j by using

$$SR_j = O_{\bullet,j} * SL. \quad (1)$$

Speed of the Persons. A triangular distribution of (0.5,1.0,1.5) m/s is assumed for the speed of the persons.

Distribution of the Persons at the Destinations. The distribution of the persons at a destination cannot be measured. Optimization calculations were used to determine the percentage distribution of entering persons with respect to the reachable destinations.

Path network with Path Lengths. The persons' paths and the corresponding path lengths were determined by using the local conditions and were quantified.

4.3 Online Data for Model Initialization

At the start of the forecast, the simulation model has to be initialized. Two problems arose when initializing the model:

- Not every model variable can be assigned a corresponding measured value from the real system.
- The measured valued contain measuring errors.

Defining k as the interval that corresponds to the current time and at which the simulation should start, then only the following variables (see table 1) can be measured or computed:

Table 1: Measurable and Computable Model Variables

Model variables	Measurable	Computable
$\mathbf{I_{j,k}}$	yes	-
$\mathbf{O_{j,k}}$	yes	-
$\mathbf{C_k}$	No	Yes
$\mathbf{CS_k}$	No	No
$\mathbf{CP_k}$	No	No
$\mathbf{CW_k}$	No	No
$\mathbf{CA_k}$	No	No

Only two model variables can be measured directly, the number of incoming and the number of exiting persons for any gate in last interval.

That means, the number of persons in the system can be computed based on both measurable variables.

If the following are designated thusly:

- $I_{\bullet j}$ the number of persons that have entered the concourse through any door during interval j and
- $O_{\bullet j}$ the number persons that have exited the concourse through any door during interval j,

then the number of persons C_k still in the system at the end of interval k can be calculated by using

$$C_k = \sum_{j=1}^{k} I_{\bullet j} - \sum_{j=1}^{k} O_{\bullet j} \quad (2)$$

The values necessary for the model variables of number of persons from the service area CS_k, number of persons on the paths CP_k, number of persons in the waiting area CW_k and number of persons in the service area CA_k can be neither computed nor measured.

Section 5 demonstrates how the assumptions for these values in initialization were arrived at.

4.4 Errors in Measurement

All measured valued are erroneous. In this case-study it was possible to estimate an error function for each of the measuring devices and all measured values were respectively adjusted before they were stored in the data base. Hence, all measuring data can be considered correct.

5 IMPLEMENTATION IN SLX

The simulation system SLX was used to implement the methods for initialization described in section 3.

5.1 Initialization with Model Generation

In this form of initialization, the model variables are initialized for the interval k with the available measured data or the computable data. The remaining model variables are initialized based on model-dependent historical observations or assumptions. The following was assumed for variables that are not measurable and not directly computable:

- CP_k – number of persons on the paths equals zero
- CS_k – number of persons in the service area equals the number of persons in the system C_k
- CA_k - total number of persons interacting with the services equals 1
- CW_k - number of persons waiting equals the difference of CS_k - CA_k

The forecast simulation starts on the basis of these model initialization parameters.

5.2 Initialization with Parent Model Synchronization

The "requested" synchronization approach is used for synchronization. The update rhythm of the measured data necessitates this. The update interval, i.e. the time between two updates, is 5 minutes, thus ruling out permanent synchronization approach for this case study.

The parent model receives a request for synchronization at the real time T_{NOW}. This time T_{NOW} marks the end of the interval k. The parent model reflects the current state of the interval k-1. The current simulation time in the parent model is t_{i-1}. The simulations starts and uses as input data the measured values for the number of persons that entered through corresponding entrances in the interval k. The simulation is terminated when the current simulation time t has reached the value of T_{NOW}.

Taking all the measured data and known conditions in the real system into account, the values of variables computed by means of simulation then have to be corrected if necessary. The correction for the variable of the number of persons in the service area is shown can be cited as an example.

If the following are designated thusly:

- C_k^{SIM} the simulated number of persons in the system for interval k,
- C_k^{MEAS} the number of persons in the system for interval k computed from the measured data,
- SL the service level (probability that a person uses this service)
- CS_k^{SIM} the simulated number of persons on the service area for interval k and
- CS_k^{CORR} the corrected number of persons in the service area for interval k,

then the corrected number can be calculated by using Equation 3:

$$CS_k^{CORR} = CS_k^{SIM} + SL * \left(C_k^{SIM} - C_k^{MEAS}\right) \quad (3)$$

The simulated number corrected by the proportional difference from the simulated total number minus the computed total number of persons for the interval k produces the corrected number used for initialization. If, in the simulation, "too many" persons in the system were calculated, then, when the inflow of persons is correctly simulated, the outflow of persons has to be corrected. In this case, persons are removed from the service area. In the case that are not enough persons are in the system, additional persons are inserted to the service area.

Analogous corrections were made for the other variables. These correction algorithms were implemented in parent model.

Since the simulation system SLX does not have the capability to generate an external child model, the child model was emulated within the parent model. The parent model is operated with a combination of two different time advance mechanisms.

The SLX simulation system has options for storing the current model state (checkpoint) and resetting a model to a stored state (restore). The first applications of checkpoint/restore technology go back to the beginnings of simulation when the simulation runs lasted much longer than hardware and operating systems could run without interruption or crashes. When a simulation was interrupted, an available checkpoint could be reverted to.

This technology is now used to emulate a child model. After the parent model has been synchronized, a checkpoint is set at the time and the parent model switches into the model range of the child model. The simulation runs is executed until the end of the forecast period, the results are evaluated and the model is reset to the model state stored in the checkpoint. If necessary, several independent runs are executed.

After the forecasts with the child model have been completed, the model range of the parent model is reset. The parent model sleeps until it receives another request for synchronization. Figure 9 illustrates the main flow inside the SLX-Implementation based on UML activity diagrams.

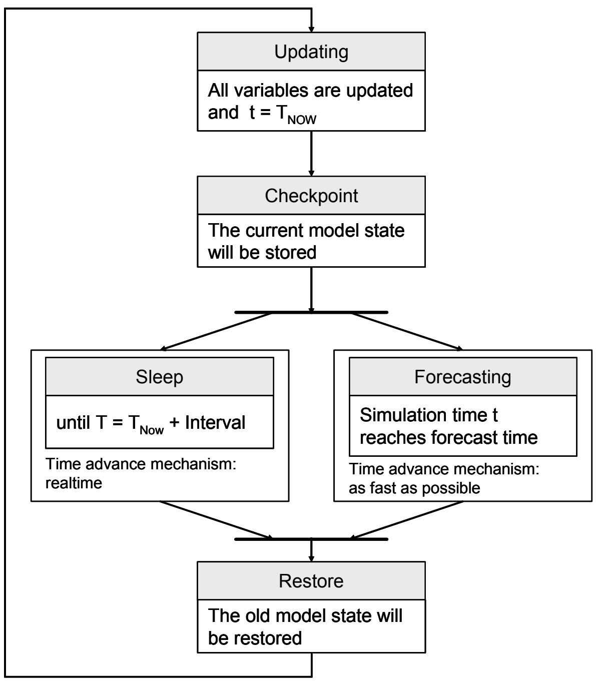

Figure 9: Main Loop in the SLX-Implementation

The emulation approach requires the simulation-based forecasts with child model to be terminated before the parent model receives a new request for synchronization.

Figure 10 illustrates the real and the simulated progression of the system variables of persons in the service area. The thick line describes the progression up to the start of a forecast. The thin lines indicate different forecast progressions. The differences result from the application of different random numbers for the forecast.

6 SUMMARY AND OUTLOOK

Online simulation provides a way to improve the quality of online controls and simulation-based early warning systems. The online simulation models must reflect the current state of the system to be simulated. On the one hand,

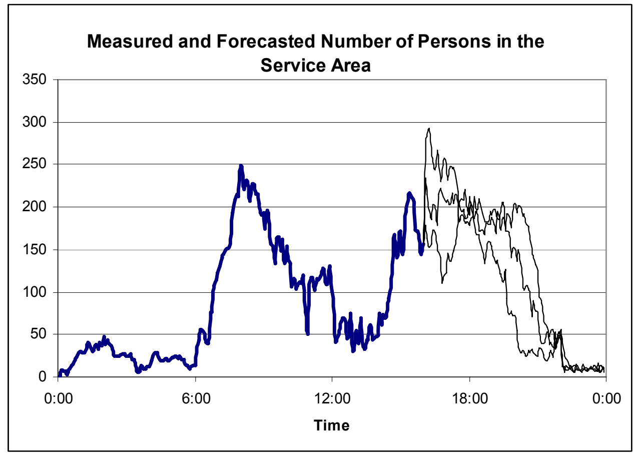

Figure 10: Measured and Forecasted Number of Persons in the Service Area

in many cases, not all the variables used in the models can be initialized with measurable or computable values from the real system. On the other hand, the measured values contain measuring errors or the measured values are not up-to-date. Since transient phases have to be omitted to adjust the statistics, incorrect initialization of the online models directly affects the findings derived. Possibilities have to be sought to keep the deficiencies during initialization as slight as possible.

To realize this goal, the synchronization approach is recommended during initialization. This approach incorporates the "knowledge" implicit in a model and allows qualitatively better initialization. This advantage comes at the cost of more effort for implementation.

A similar approach of the implementation shown in this paper can be transferred to more complex systems also. However, no generally accepted correction instruction can be specified to do this. The correction are specific to conditions of the model.

Other studies will examine the problem of allowing for measuring errors in online data as well as the efficient organization of the management of parallel child models.

REFERENCES

Chong, C. S., A. I. Sivakumar and R. Gay. 2002. Design, Development and Application of an Object Oriented Simulation Toolkit for Real-Time Semiconductor Manufacturing Scheduling. In *Proceedings of the 2002 Winter Simulation Conference*, E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds. pp. 1849-56.

Davis, W. J. 1998. On-Line Simulation: Need and Evoling Research Requirements. In Handbook of Simulation. Edsited by Jerry Banks. John Wiley & Sons, Inc. pp-465-516.

Glinsky, E. and G. Wainer. 2002. Performance Analysis of Real-Time DEVS Models. In *Proceedings of the 2002 Winter Simulation Conference*, E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds. pp. 588-94

Gupta, A. K., A. I. Sivakumar and S. Sarawgi. 2002. Shop Floor Scheduling with Simulation Based Proactive Decision Support. In *Proceedings of the 2002 Winter Simulation Conference*, E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds. pp. 1897-1902

Fowler, J. W. and O. Rose. 2004. Grand Challenges in Modeling and Simulation of Complex Manufacturing Systems. SIMULATION, Sep 2004; 80: 469 - 476.

Harmonosky, C. 1995. Simulation-based real-time scheduling: review of recent developments In *Proceedings of the 1995 Winter Simulation Conference*, C. Alexopolos, K. Lang, W. R. Lilegdon, and D. Goldsman, eds. pp. 220 – 225

Hanisch, A., J. Tolujew, K. Richter and T. Schulze. 2003. Online Simulation of Pedestrian Flow in Public Buildings. In *Proceedings of the 2003 Winter Simulation Conference*, S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds.. pp. 1635-41

Katz, D. and S. Mannivannan. 1993. Exception Management on Shop Floor Using Online Simulation. In *Proceedings of the 1993 Winter Simulation Conference*, G. W. Evens, M. Mollaghaseme, E. C. Russel, W. E. Biles, eds. pp. 888-896

Manivannan, S. and J. Banks. 1991. Real-Time Control of a Manufacturing Cell Using Knowledge-Based Simulation. In *Proceedings of the 1991 Winter Simulation Conference*, ed. B. Nelson, W.D. Kelton, and G. Klark, pp 251-60.

Mazur, F., R. Chrobok, S.F. Hafstein, A. Pottmeier and M. Schreckenberg. 2004. Future of Traffic Information - Online-Simulation of a Large Scale Freeway Network. In: *Proceedings of IADIS International Conference WWW/Internet 2004*. Volume 1.,pp. 665-672.

Sivakumar, A. I.. 1999. Optimization of Cycle Time and Utilization in Semiconductor Test Manufacturing Using Simulation-Based, On-Line, Near-Real-Time Scheduling System. In *Proceedings of the 1999 Winter Simulation Conference*. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds. pp. 727-35

AUTHOR BIOGRAPHIES

JURI TOLUJEW is a project manager in the Department for Logistics Systems and Networks at the Fraunhofer Institute for Factory Operation and Automation in Magdeburg, Germany. He received his Ph.D. in automation engineering in 1979from the University of Riga. He received his habil. degree in computer science in 2001 from the University of Magdeburg. His research interests include simulation-based analysis of production und logistics systems, protocol-based methods for analyzing processes in real and simulated systems as well as mesoscopic approaches in simulation. He is an active member of ASIM, the German simulation association. His email address is `<juri.tolujew@iff.fraunhofer.de>.`

ANDRÉ HANISCH is a research engineer in the Department for Logistics Systems and Networks at the Fraunhofer Institute for Factory Operation and Automation in Magdeburg, Germany. His Diplom thesis completed in January, 2002 dealt with “Groupware Components for Collaborative Work in a Web-based Mining Simulation Center”. His e-mail address is `<andre.hanisch@iff.fraunhofer.de>`.

THOMAS SCHULZE is an associate professor in the School of Computer Science at the Otto von Guericke-University, Magdeburg, Germany. He received his Ph.D. in civil engineering in 1979 and his habil. degree for computer science in 1991 from the University of Magdeburg. His research interests include modeling methodology, public systems modeling, manufacturing simulation, and distributed simulation with HLA. He is an active member in ASIM, the German simulation association. His email and web address are `<tom@iti.cs.uni-magdeburg.de>` and `<http://www-wi.cs.uni-magdeburg.de>`.