Computer Simulation Technology and Demonstration

S. Schafrik & M. Karmis
Virginia Center for Coal and Energy Research, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA

Z. Agioutantis
Technical University of Crete, Hania, Greece

T. Henderson
Henderson Associates, Bluefield, West Virginia, USA

ABSTRACT: The algorithms for the mathematical modeling to predict productivity of underground room-and-pillar mining systems are well known and documented. These algorithms consider the time-varying relationships between mining equipment for a given geometry of operations as well as other constraints. A user-friendly visual simulation computer tool for the Windows environment is demonstrated and shows how mine production can be scheduled. The simulator, which is easily customized and utilized by the field engineer, can help mine operators plan the optimum mining sequence for different mine geometries and equipment layout. Program output includes monitoring of shift data, equipment utilization indices, etc. Delays, such as equipment failures, maintenance and chain-of-supply, are also included in the simulations. This simulation technique can be used to perform simulations in any production environment.

1 INTRODUCTION

Imitating the operations of real-life systems or processes is the main purpose of computer simulation. Operational scenarios can be tested and evaluated without the need or expense of physical experimentation. Applications have been developed to simulate the space and time relationships between mining equipment, mainly in connection with transport systems (Topuz et al. 1989, Zhao & Suboleski 1987, Ramachandran 1983). Zhao & Suboleski give a detailed account of the existing mine simulators available in the late 1980s, including CONSIM (Topuz et al. 1989), FACESIM (Prelaz et al. 1968), and FRAPS (Haycocks et al. 1984) developed at Virginia Tech and UGMHS developed at Penn State. Additionally, simulators with graphics or animation capabilities such as MPASS-2 are also mentioned. SAM, the simulator developed by Zhao (Zhao & Suboleski, 1987) can be added to this list as well as FACEPROD, a simulator developed by Hollar (Hollar, pers. Comm.).

These dedicated simulation packages were developed in general purpose programming languages such as Fortran, Pascal and Basic. For the most part these programs are written to use input information that describes the physical environment and the equipment capabilities, then use mathematical algorithms, vary the input data, and calculate production rates, tons produced, etc. They generally have one or more reporting options that can output information in standardized reports.

Simulation scripts written in general purpose simulation languages such as GPSS, GPSS/H, Automod, etc. (Vagenas 1999, Sturgul 1999) have been applied towards the development of discrete event simulation software packages for both underground and open-pit mining operations. The GPSS (General Purpose Simulation System) language and the event driven version, GPSS/H, are some of the best-documented discrete event simulation languages for use in mining situations. Sturgul (2000) wrote a book on the topic of applying GPSS/H to mine design situations. Examples and case studies in this book as well as numerous publications by John Sturgul demonstrate the applicability and ease of use of GPSS/H to mining and minerals engineering simulation problems. Bethlehem Steel designed a belt simulation named BETHBELT-1 written in GASP V, a programming language designed for discrete-event simulation. Another belt simulation, developed to handle 25 belts and 12 loading zones, was written in the simulation language known as PL/1.

2 SIMULATOR OBJECTIVES AND IMPLEMENTATIONS

This paper focuses on WebConSim a program originally developed by the authors in 1999 (Schafrik et al. 2001). Development of the program has continued and many changes to coding and concept have been implemented. WebConSim was developed to be a Frame-Based Expert System (Durkin 1994) for the prediction and modeling of underground coal-mine equipment over time. A frame is a computer concept that represents real objects, such as individual pieces of mining equipment. The development of object-oriented programming practices has helped foster new functionality in frame-based expert systems. The equipment is not controlled by equations or queues that govern cycle or service time. Instead, the equipment frame examines the simulation environment and makes decisions about its own state based on the environment.

Object-oriented programming also allows a program to be written using a client / server architecture for collaborative software systems. WebConSim was designed to use the power of client / server architecture in order to allow access to the simulator irrespective of the type of personal computer utilized by the end user. The actual simulator resides on the server and any kind of client can communicate user information to the server, i.e. Microsoft Windows, Palm Pilots, Unix, Apple Macintosh, etc. Input data is stored in a database, which also resides on the server. A field engineer, planning engineer, mine management and holding company can all, therefore, access the same information simultaneously. To implement this design, three separate modules were developed: The front-end module, the database module and the simulator module.

2.1 *Front-End Design*

One of the benefits of using a web-based front-end, is that a client / server architecture can be utilized, where the client communicates with the server by sending normal http requests. This is easily implemented using a technology such as Active Server Pages (ASP). ASP allows a single database to be located on a server and all web content to be generated when the browser asks for it. ASP is compatible with all popular Internet browsers that have been made in the past seven years because the browser views normal web content and all ASP processing occurs on the server, rather than on the client. The other major advantage to the web-based deployment is that a software upgrade need only be installed on the server, not on the many clients. The simulator module can only be hosted by Microsoft Windows 9X or NT computer. This computer must have Microsoft ActiveX Data Objects (ADO) version 3.5 or later and Microsoft Internet Information Services version 4 or later with Active Server Pages option installed (Personal Web Server on Microsoft Windows 9X).

The simulator itself is not an executable application; it is an ActiveX Dynamic Link Library (DLL) that is executed from an application. An important advantage of using the new ActiveX technology is that it provides for encapsulation and inheritance. Thus, changes to the simulator can be made without the need of source code or recompiling the simulator. This includes the addition of equipment that is not accounted for at design time. Once the operational logic has been worked out, adding new equipment to the simulator can be accomplished using inheritance. Each user has, in effect, his or her own version of the simulator, tailored for individual, rather than general use (Schafrik et al. 2001).

The web site is designed with maintenance and action areas. In the maintenance area, the user inputs information into the simulator database. This area has features that help to automate the process, such as equipment copy and paste functions, thus reducing the need for repetitive entries. The report maintenance area contains all the output from simulations that the user has not deleted (Schafrik et al. 2001).

Currently, development work is conducted using Microsoft Visio and AutoDesk AutoCAD as a graphical front-end, rather than the web-based approach. Although the sharing of input data will be lost by using these programs, it makes the simulation environment more interactive. The target is to allow a user to link the graphical information to the simulation in a way that has not been done before. This could also allow for the user to change variables (e.g. tram speed, cut rate, capacity, etc.) while the simulation is running. Such interactivity could allow a user to make minor adjustments to a model without waiting until an entire simulation has been completed. This would also help a user to identify bottlenecks visually, rather than through review of text and data reports.

2.2 *Database Design*

The purpose of the database module is to supply information to the simulation engine. Information is stored in a Microsoft Access 2000 database and the simulator uses ADO to connect to the database (Microsoft Access was used as the development database for convenience only). The user may override the development platform and substitute the current implementation with a Microsoft SQL Server, an Oracle database, or any ODBC database. The data stored in the database is divided into the following major groups (Schafrik et al. 2001):

- Layouts
- Cut Sequences
- Waypoints
- Travel Paths
- Teams

“Layout” information concerns only the physical dimensions of the area to be mined. These dimensions do not vary over time, depth or length. The database can include more than one layout corresponding to different sections of a mine or different mines.

“Cut sequences” govern the movement of the miners and the roof bolters. The cut sequences are repeated between feeder moves allowing multiple shifts to be simulated without repetitive input. Cut sequences may be generated automatically (using separate program modules) or keyed in manually (Fig. 1).

The physical movement and location of all equipment is based on “waypoints.” A waypoint is defined as the center point at each entry / crosscut intersection. Waypoints serve as points of travel and as reference points for all mobile equipment. When these points are generated the inby, outby, left and right neighbor relationships are also calculated. Also, these points track of how much material is scheduled to be mined and / or has been mined in each one of these directions. The same information is kept regarding the length of the bolted entry in each direction. In summary, the calculation of all physical locations of moving equipment is based on the waypoints. The waypoints are stored in the database, but are generated in each simulation run based on the layout information. This is a new feature in the simulator, accommodating sections with fluctuating pillar sizes and providing more user control over the simulation.

“Travel paths” are critical to the overall effectiveness of the system. The database stores the travel paths as defined with reference to waypoints. The distance and number of turns is a property of the path. For any single to-and-from pair, there must be more than one entry in the paths section of the database, unless only one piece of equipment can travel between the two points at any given time. For instance, if a shuttle car has been loaded by a miner at waypoint 7 and would like to tram to the feeder located at waypoint 14, it must look for an available path entry from 7 to 14, then the amount of time to make the journey is calculated (Fig. 2).

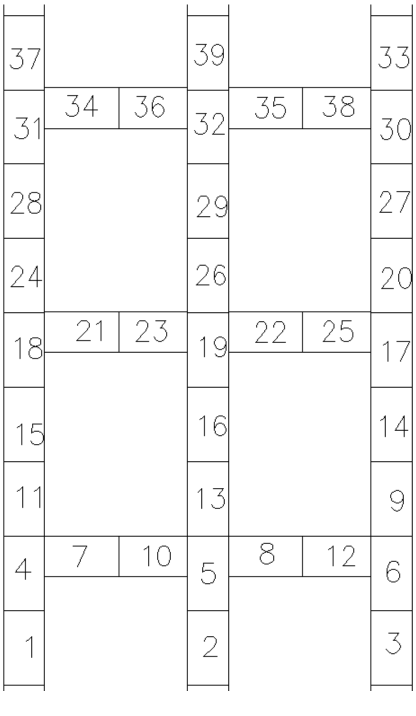

Figure 1. Example of a cut sequence.

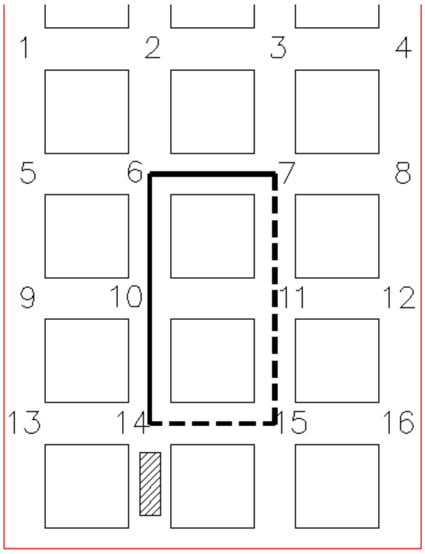

Figure 2: Waypoints and travel paths.

“Teams” are mining crews and may include any combination of mining equipment. This section of the database is the most complex. The design is intended to reduce the query time and complexity, allowing information to be easily duplicated while the user is compiling a team. When the simulator is told what team to use, it obtains all the equipment by querying the equipment tables for the appropriate team identifier. Storage of teams in this manner increases the ease of copying equipment from one team to another.

There are many pieces of information that are not unique to a specific piece of equipment (e.g. mean time between failures, mean time to repair, statistics type, location, and initial state). This data is critical for all equipment for the simulator to operate. The statistics type can be deterministic or stochastic and is directly related to the rate profile of the equipment. The profile includes the average, standard deviation, minimum and maximum and need only store the proper data for the chosen statistics type.

2.3 *Simulator Logic, Design and Improvements*

The simulator implements an expert system engine, where the focus is not on scenario but on equipment operations. This focus makes it truly event-driven. Every equipment unit maintains its own Time to Next Event (TTNE), i.e. the time to complete the current task and re-evaluate its state. The simulation engine is interested in three things:

– the original state of equipment,
– the final location of equipment, and
– the capabilities of equipment.

From these three pieces of data a complete simulation to and from any time point can be accurately performed Specifically, this expert system approach is implemented using equipment states. While in a

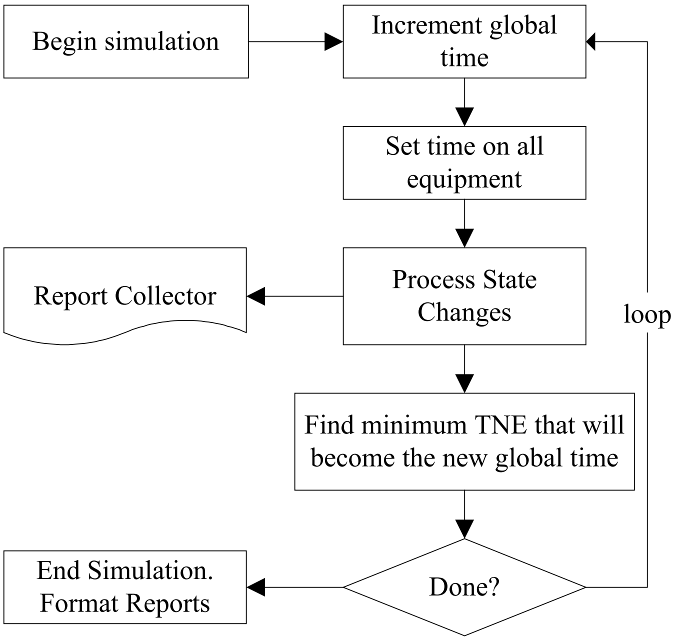
Begin simulation
Increment global time
Set time on all equipment
Report Collector
Process State Changes
loop
Find minimum TNE that will become the new global time
End Simulation. Format Reports
Done?

Figure 3. Flow of simulation.

state, there is a specific period that the piece of equipment will remain in that state. Its TTNE is the current time plus the current operation time. Once a state has been completed the equipment needs to make a decision regarding the next action to begin. This decision is a function of the current state, the time in the current state and what the next job is. Any kind of equipment can easily be added by allowing for appropriate state changes. The simulator operates as shown in Figure 3. It is important to note that the main simulator loop sets a global time and finds the next time to set. The setting of the global time is the point in the loop where the processing is actually done. When the time is set on a piece of equipment, the equipment checks if its TTNE (or time to state change) is equal to the current global time. If the times are the same, then the equipment decides on what its new current state is and how long it will be in this state. Once this has been completed for all equipment the simulator engine will "ask" all the equipment for their TTNE, then it will calculate the minimum of these values and use that as the new global time. This allows for multiple pieces of equipment to be running different things concurrently.

The type of simulation described is a new approach. It uses relatively new programming technologies to allow for multiple smart objects to be located in an artificially confined area. Each object decides on the optimum way to accomplish its plan and then executes that procedure.

Each equipment parameter (i.e. tramming rate, breakdown rate, etc.) can be evaluated based on statistical information available for the specific piece of equipment. Thus, equipment parameters can be estimated either deterministically or stochastically. Stochastic distributions currently support normal distribution (e.g. for tramming rates), exponential distribution (i.e. for breakdowns and repairs), uniform distribution and user-defined distributions, where the user can enter a measured cumulative distribution function in a table form to be used by the simulator.

Deterministic analysis uses only an average for the variable. Uniform analysis uses an upper and lower bound for the variable. Normal distribution uses average and standard deviation, while exponential distribution uses mean rate. User-defined analysis allows the user to create ranges of values based on a range of probabilities (e.g. from 0% to 25% is 3, from 26% to 75% is 5, 76% to 100% is 10). While determining the TTNE, each piece of equipment will use the statistical analysis for the specific operation. For example, the tram rate is an input variable for most equipment, where the product of the tram distance * tram rate yields the time to complete a tramming operation. The tram rate in the above relationship will be determined based on the statistics set for the specific piece of equipment.

Breakdown times, delay times and operating times are calculated based on mean time between failures and mean time to repair. These are assumed to be exponentially distributed because they are inter-arrival times. This functionality cannot be overridden.

A delay function has been added to the simulator. This function allows for outside interference to affect the mining process. Currently, there are four different types of delays that can be built into the system: physical conditions, outby operations, belts, and methane. The user has the option of selecting which delays affect which equipment. For instance, the user can set a shuttle car to be unaffected by outby operations but affected by all other delays. Another good example is the roof bolter. Typically roof bolters are more affected by lack of supplies from outby operations than shuttle cars and other equipment. This new feature allows a user to run simulations under imperfect conditions to get more realistic results.

2.4 *Reporting*

The simulator itself does not generate the reports. The process, which requests the simulation, carries out this function. There are currently four reports that are generated:

- standard report
- multicycle report
- extended report and
- verbose report.

The standard report shows the traditional information that is expected from a mining simulator. The report works by monitoring the EndofCut and EndofBolt event in the mine object. When the cut is completed there is a summary table and a table for every piece of equipment that is generated for that cut, except the roof bolter. The roof bolter's table is gathered at the EndofBolt event. The summary table includes information such as tons mined and tons per minute. The individual equipment tables show the amount of time the equipment was in each state and other equipment-specific collected information, for example, amount cut (miner) and number of trips to the miner (shuttle car).

The standard report also shows summary information for equipment performance in the mine. It was intended as the report that is typically used, especially for new equipment and mine combinations. When interesting results are shown on this report, the extended or verbose report can be used to ascertain the source of the anomalies. Those reports can also be used to identify possible modifications to the configuration in the mine design.

The multicycle report is an extension of the standard report. It functions in the same manner, except that the tables presented aren't built during the simulation. The multicycle report is identical in form to the standard report, but shows summary information over the course of the many simulation cycles. It is intended to allow the user to run the same simulation for many cycles and then review the results. For equipment that is using non-deterministic statistical analysis, this type of report will output a more accurate representation. After the last simulation cycle the report generator creates a summary of the individual tables.

The extended report shows the state of the mine at every simulation cycle. It creates an instance of the simulator's application object and waits for the EndGetTTNE event to begin. When this event begins, the extended report gathers the current time, current cut and bolt number. Then it cycles through all the equipment and collects the current state, last state, location and TTNE. This report is useful in examining the state of the all the equipment and their interactions throughout the simulation. This report can also be used as input to a visualization of the simulation. Because this report shows a line only when there is a change in the simulation, a visualization program can interpolate between locations and calculate the state of the equipment between the simulation cycles. This report is also useful in identifying equipment that is being under-utilized because it is frequently either waiting or queuing.

The verbose report watches the report object for when a new event is added. When the event is added the object adding the event is passed to the verbose report object. The verbose object will check the mine object to get the current cut number. It then collects the current state, previous state, id, description, current time, TTNE and location from the piece of equipment. This report is very valuable in understanding the equipment's reasoning for altering states. This report produces a complete log of the equipment for the simulation time. WebConSim is the only simulator that is capable of delivering such a report.

3 EXAMPLE OF SIMULATION USAGE

A case study was completed using the simulator as an analysis and prediction tool for an existing coal mine located in the Central Appalachian Coal fields. Simulator input was based on existing mining reports and a model was generated that can be used to predict mine production and scheduling with the existing set of mine parameters. An alternative mining scenario was executed by modifying the equipment configuration. The productivity and other changes to the mining cycle were then calculated. In this particular case, the equipment change also occurred in the mine and the same production reports were compared to the changes predicted by the simulator.

The coalmine used for this study has the following basic parameters:

- A general mining height of 119cm with a total rock parting of approximately 56cm, and high variation of both.
- The section is a 7-entry section working two shifts with 8 men per shift.
- The pillar size averages 21.3m x 21.3m with 5.5m entries (Figs 1-2).
- The roof bolter is a double boom bolter roughly equivalent to the Fletcher Roof Ranger II.
- The continuous miner is capable of 12.1m cuts and has a 96.5cm cutting drum.
- The mine averages approximately 55 linear meters of advance, but this number is highly dependent on shift, working conditions, and equipment.

While building the model for use in the simulator, the information above was used as general parameters and the haulage of the coal at the working section was done using three battery-powered ram-cars. Production and delay reports for approximately one month were used, in conjunction with factory equipment specifications, to create a model. This model was run using the stochastic capabilities of the simulator in order to match the simulator output to the mine's historical output.

The information in both the verbose and summary report from the simulator show the same trend. The simulator's output matched the production reports in an acceptable manner. The information used for the teams and the layouts of the mine was used as a model. Analysis of output from the simulator showed several different bottlenecks in the mining processes:

- Ram-car battery-life and change-out time caused the most production delays.

- In many shifts it was common (i.e. for a long period of the shift) for only two cars to be in service at any point in time.
- In addition to production delays caused by the haulage system, there were significant production delays caused by mined areas that had not been supported by the roof bolter.

Using the model created with the equipment configuration and the mine layout a change to the haulage system was made in the simulation. The three battery haulage cars were replaced with two electric powered shuttle cars. The shuttle cars are rated for approximately a 5% capacity increase over the ram cars. Simulations were run using the two cars and showed a similar production rate using one less piece of equipment. In the latter case it was shown that the roof bolter caused the bottleneck in the production chain. Because there are seven rooms to be bolted and three pieces of equipment to create more rooms, roof bolting was falling behind production. Without a safe area to work and mine coal, there can be no production.

In this case study, the same production data was used to verify the output from the simulator. It was found that the simulator results were equivalent to the real production results.

4 CONCLUSIONS

There is a need for tools that can aid in the continuing increase in productivity for current and future mining conditions. The simulator presented, WebConSim, fulfills two major objectives, to utilize standardized computing technology and to reflect current and future room-and-pillar coal mining practices. WebConSim fulfills the first objective, because it is built on a frame-based expert system architecture, implemented as a standard client / server application. The frame-based expert system architecture allows for the system to be configured for current mining operations and to easily adapt to new mining practices, thus fulfilling the second objective.

The client / server implementation architecture, allows for the simulator to be interfaced through a web-based front-end and work is being done for non-web-based interaction. Two equipment types are missing from the system that are widely used in the coalmines, the miner-bolter and bridge conveyors. With the addition of these new equipment types, a form of error-checking must also be added to insure that an equipment group has the proper equipment to mine. With these additions the simulator itself will be made more universal.

In the example presented it was shown that a model of existing mine production can be made and a change to that model can then be analyzed. In this case, predicted results from the simulator can be used for planning purposes. This technique can be used to aid a mine operator in making decisions regarding equipment and mining scheduling. As continued development and use of the simulator progresses, it can more accurately reflect current and future mining practices.

The future development of the simulator will focus on increasing its interactive capabilities during simulation of mining operations. “On-the-fly” changes to the model will be possible and will give the user a better understanding of the simulator’s methodology. With greater understanding, the simulator will no longer appear to be a “black-box” calculator, simply generating numbers.

REFERENCES

Durkin, J. 1994. *Expert Systems Design and Development.* New York: Macmillan. 800.

Haycocks, C., Kumar. A., Unal, A. & Osei-Agyemana, S. 1984. Interactive Simulation for Room and Pillar Miners. In Institute of Mining and Metallurgy (ed.), *18th APCOM, London England,* 5591-597.

Hollar, P. 2000. Personal Communication, February, 2000.

Prelaz, L., Bucklen, E., Suboleski, S. & Lucas, R. 1968. Computer Applications in Underground Mining Systems. R&D Report No. 37, Vol. I. Washington: Office of Coal Research.

Ramachandran, D. 1983. *A Simulation Model for Continuous Mining Systems, M.S. thesis.* Blacksburg, Virginia: VPI & SU.

Schafrik, S., M. Karmis & Z. Agioutantis. 2001. A Novel, Web-Based, Continuous Mining Simulator. *Transactions, Society for Mining, Metallurgy and Exploration, Inc.* 310: 69-74.

Sturgul, J. R. 1999. Discrete Mine System Simulation in the United States. *International Journal of Surface Mining, Reclmation and Environment* 13: 37-41.

Sturgul, J. R. 2000. *Mine Design: Examples using Simulation.* Littleton, Colorado: Society of Mining, Metallurgy, and Exploration, Inc. 367.

Topuz, E., Nasuf, E. & Ramachandran, D. 1989. *Consim: An Interactive Micro-computer Program for Continuous Mining Systems., User Manual, Second Revision.* Blacksburg, Virginia: VPI & SU.

Vagenas, N. 1999. Applications of Discrete-Event Simulation in Canadian Mining Operations in the Nineties. *International Journal of Surface Mining, Reclamation and Environment* 13: 77-78.

Zhao, R. & Suboleski, S. 1987. Graphical Simulation of Continuous Miner Production Systems. *APCOM 87: Proceedings of the 12th International Symposium on the Application of Computers and Mathematics in the Mineral Industry.*