Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

MODELING A BULK MANUFACTURING SYSTEM USING EXTEND™

Andrew J. Siprelle
David J. Parsons

Simulation Dynamics, Inc.
506 West Broadway Avenue
Maryville, Tennessee 37801-4712, U.S.A.

ABSTRACT

While simulation is readily used for traditional "pieces/parts" manufacturing problems, it is also highly effective in solving problems in the world of bulk (dry) product or fluid manufacture. The nature of bulk or fluid manufacturing problems calls for a unique simulation architecture which is adaptable to many different applications. This presentation shows how the simulation package Extend by Imagine That, Inc. of San Jose CA can be used to address such problems. A model of a soap manufacturing operation illustrates the use of this approach for determining the operational impact of expansion projects requiring significant capital while also allowing the user to determine optimum schedules for plant operation. In the long term, the model will likely be used on an ongoing basis to analyze production sequences.

1 INTRODUCTION

The world of consumer products manufacturing is increasingly under pressure to produce products at a lower cost while at the same time offering more variety. With the growing trend of large retailers to "tie-in" their manufacturing suppliers to their distribution system, the burden of meeting this demand for more variety passes to the manufacturing facilities. One needs to anticipate an increasingly flexible operation which can provide a variety of product on demand. Such an operation typically employs technology to enable flexible flow routing and control for quickly changing between formulas or packaging. These new capabilities present challenges in how to design, operate and manage an increasingly complex production network.

Simulation can be readily applied to address these complexities. Simulation has been most applicable to problems in the "pieces/parts" industry; e.g. production of car parts, widgets, etc. The "world view" of these systems can be summarized by a model of entities flowing through a queuing system. Unfortunately, attempts to aggregate chunks of bulk product into entities yield models that either run too slow or are too inaccurate to use. Another approach is to use combined discrete/continuous languages which emphasize writing differential equations and additional code. However, a better approach is possibly using an architecture designed to address the nature of bulk/fluid production. This architecture is developed as part of an Extend library, which can be used to build a wide variety of bulk/fluid flow models.

2 THE SOAP PRODUCTION SYSTEM AND THE USE OF A MODEL

The soap manufacturing system consists of 4 production lines as shown in Figure 1 below:

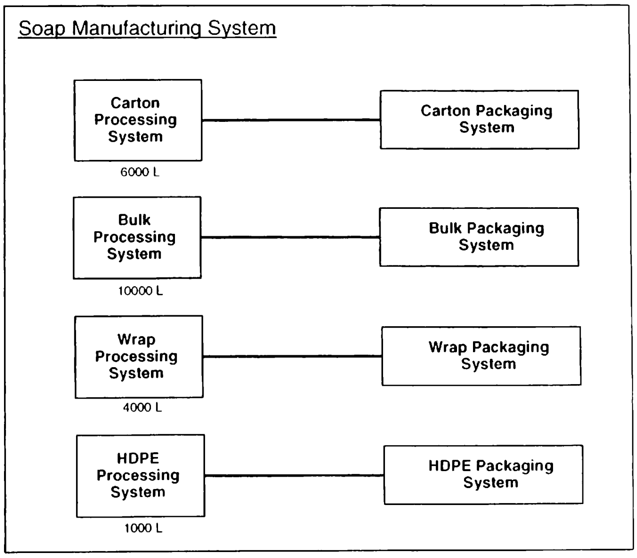


Figure 1: The Soap Manufacturing System Processes

While lines run similar formulas each line processes and packages these formulas independently.

Each line runs 40 to 100 different products. To produce a campaign of product, a load of the correct formula is prepared by the Processing System. The size of this load is set as a predetermined quantity for each formula. This quantity is described in terms of the percentage of the capacity of the Mix Phase 2 Tank. The load size is designed to be an appropriate amount to run to balance production yet minimize changeovers. It is also designed such that additions are simply measured.

To start a load, crews add various amounts of material to one or both Mix Phase 1 tanks. Note that for a few SKUs, the initial load actually skips the Mix Phase 1 altogether. Material for such loads is added directly to the Mix Phase 2 Tank. If the load material is Phase 1 mixed, the two Mix Phase 1 tanks are plugged while the material undergoes an agitation phase in one or both tanks.

Once homogenization is achieved, the load is pumped to the Mix Phase 2 Tank, where addition of material and agitation occurs again. Once this is done, the material in the load circulates through heat exchangers. Once done with recirculation, a 3-way valve is opened for pumping to one of the Product Overflow Tanks. Note that a few SKUs skip the heat exchange process and go directly to overflow. The entire time required to prepare a batch and fill an Overflow tank ranges from 0.5 to 5 hours. Once the load is placed in overflow, a quick quality analysis is performed, and then the load is ready to be fed into the Packaging operation. The 2 Oveflow Tanks each provide overflow for distinct loads.

All of the tanks in this operation fill, agitate, and feed using the same control logic. This simple logic is shown in Figure 2 below:

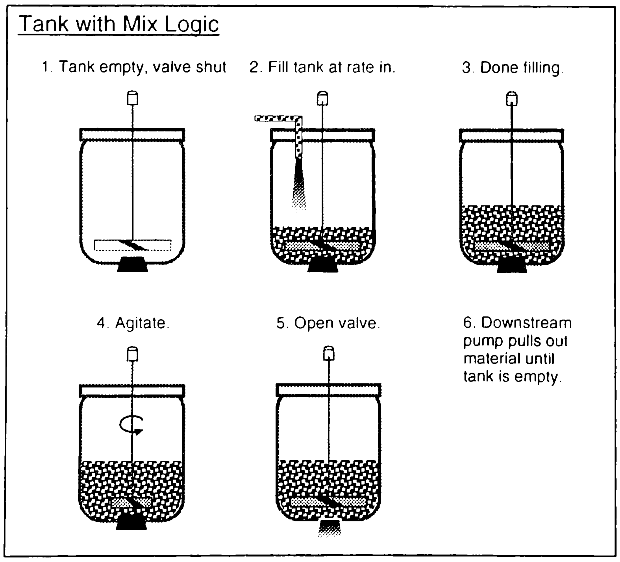


Figure 2: Tank Mix Logic Diagram

Packaging pulls the material from the Overflow Tanks continuously at the rate of downstream production; i.e., material resides in Overflow until the load is completely packaged. There are various internal operations within packaging which have very small in-process buffering. Therefore the primary buffer exists at the front of the Packaging process, in the Processing Overflow Tanks.

The rate at which product is pulled by packaging is different for each product. The calculated "pull" rate is based on fill volume per packaged unit and line speed in packaged units per minute.

A schedule for these steps must be designed to satisfy demand and make sure both materials and equipment such as tanks and flow lines will be available for different formulas of soap to be processed. Besides needing to ensure that different formulas are kept separate, strict regimens must be followed to sanitize the pipes and hoses that the fluids flow through. All of these concerns make for puzzling problems of flow routing and scheduling.

Some of the crucial questions that need answering are as follows:

- Can the processing and packaging equipment keep up with demand?
- Is it wise to spend capital for additional equipment?
- How should labor be scheduled processing and packaging?
- What is the best sequence for producing the various formulas?

To faithfully characterize the soap manufacturing system in a simulation model, the critical activities that need to be captured are as follows:

- Choosing routes and tanks for input and filling them.
- Choosing routes and tanks for output and feeding from them.
- Shutting off flow due to schedules, maintenance, or changeovers.
- Setting rates for flow between tanks.
- Delaying the release of material due to reaction times.

To successfully answer the questions posed, one could use the model to modify tank sizes, number of tanks, and various production rules and see what effect these changes have on throughput and buffer utilization.

3 THE CONTINUOUS NATURE OF THE PROBLEM

During the course of producing bulk material, a buffer undergoes continuous changes in level. At some point, the buffer becomes full or empty because of the persistence of a positive or negative net rate for a specific duration. Calculating this duration is difficult because upstream and downstream processes change rates, or other buffers become full or empty and cause blocking or starving conditions. The solution offered by continuous languages is to closely monitor when a level reaches a target, and to fire an event when this happens. To choose a small enough time step which insures against "overshooting" the target, these packages use algorithms to integrate differential equations which are specified by the user.

While the differential equation method provides flexibility when dealing with more complex rate dynamics, some simplifying assumptions allow one to effectively address a wide class of systems in the bulk/fluid process industries. By assuming that rates change in stepped fashion, one can successfully predict when buffers become full or empty. To adopt the assumption, one must envision rates that change in square-plot fashion, and persist at constant rates for some duration. Thus any effect of "ramp-up" or "ramp-down" durations between different rates is short-lived and minimal.

Making this assumption reduces the problem to one of calculating the next full or empty state. As mentioned, the problem is not trivial due to changing upstream or downstream conditions. A language must account for changes of rates, starved upstream, and blocked downstream conditions. To do this, the system must enable a fair amount of flexibility to signal when these conditions occur.

4 EXTEND AND THE BULK FLOW ARCHITECTURE

Extend is a graphical, interactive, general purpose simulation program developed by Imagine That, Inc. It provides libraries of iconic blocks, which a user can place on a worksheet and connect together. Each block is comprised of an icon with connections, a dialog interface, help text, and specific behavior which is defined by an underlying C-like language. Because Extend comes with a built-in C compiler and dialog editor, a user can modify pre-built blocks or build new ones (Diamond, 1993). A user can also build higher-level blocks (Hierarchical Blocks) which nest other blocks.

Because code is contained in the primitive (non- hierarchical) blocks, the method of simulation can change based on the library of blocks used. For example, standard libraries (Discrete Event, Manufacturing, and BPR) supplied by Imagine That, Inc. can be used to build models which use a discrete event clock and flows of items which contain attributes. Other libraries can be used to build models with a continuous time advance architecture.

Models with complex dynamics can be created using blocks connected arbitrarily in a myriad of arrangements. To support such modularity, block behavior must be robust to a huge variety of conditions. Blocks are able to "stand on their own" and deal with high fidelity with their connected neighbors. A key to this robustness is message passing, which allows blocks to query the status of other blocks which are connected upstream and downstream.

A user can create their own architecture by passing messages through connectors, or by accessing blocks or data through global variables (Krahl, 1994). Learning how to implement various aspects of a simulation architecture is aided by having access to the source code of the supplied standard blocks and by "Make Your Own" block templates.

A crucial benefit of the Extend block architecture is that one may decide to "mix and match" various pieces of simulation architecture as needed. One may decide to create blocks which adhere to standards for posting events, but bypass the basic item architecture in favor of one that permits the modeling of rates of flow. This decision was what lead to the creation of the bulk flow blocks used in this project.

The product in this model, soap, is represented as rates of flow through individual components, and as quantities in components where product mass is accounted for such as storage or delay blocks. Only the flow rate is passed from block to block. The soap in a component (e.g., tank) at any given time is known by the model's progress in working through the schedule, and by the routing of material through components of the model. This means that the soap is treated as a homogeneous material. It cannot, for example be blended. The simulation does not attempt to model product thermodynamics, particle sizes, or product constituents.

5 BUILDING THE MODEL

The model of the soap manufacturing system makes use of primitive blocks from a bulk flow library developed in the Extend ModL language, mentioned above. The blocks have been developed over several years of modeling experience in the food industry, consumer products, etc., and were modified for this application. The bulk flow blocks address the issue of what flows in the model by communicating rate information. One connects pumps, buffers with setpoints (tanks or kettles), and path diverging/converging blocks to define production routing. Additional blocks are connected to buffers to respond to setpoint changes and enact control algorithms.

The model layout is shown below in Figure 3. Hierarchical blocks help to combine model ease of understanding and navigation with lower-level detail which is nested within the hierarchy. The user can access the lower-level sub models by double-clicking on each icon representing a given area or piece of equipment.

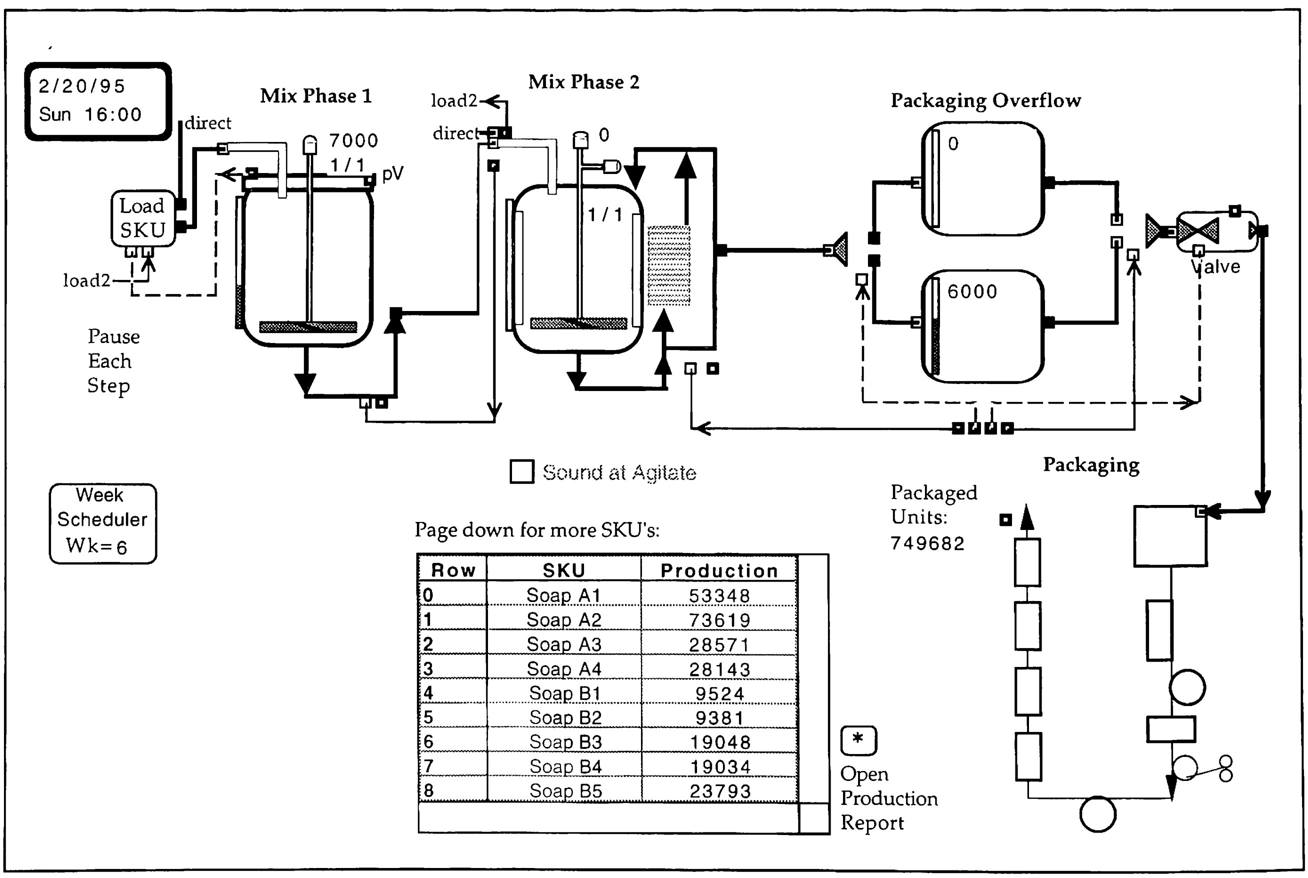
At any given time, the product moving through components in the system is managed by a model database. This means that component performance can be dependent on product characteristics such as formula type and packaging requirements. This information gets to blocks through the production schedule in the model database rather than being passed through block connectors. A spreadsheet is used to prepare the data, or data can be changed directly in the model using blocks which present views of database tables. A view of a typical database table is shown on the following page in Figure 4.

6 RESULTS OF THE SIMULATION

Using the simulation model has shown that the initial equipment installation for the soap manufacturing system is not adequate. Running the model indicated that the initial shift schedule envisioned for some lines was not sufficient to achieve required production. The model also demonstrated that a swing tank would be required to resolve a blocking problem on one of the lines. The model was then used to generate optimal schedules for operating the plant with the current production demands. Generating these schedules was easily accomplished by someone who is not familiar with simulation modeling, but needed to generate potential operating schedules for the production lines as demand for different soap products changes.

7 CONCLUSION

Accurate modeling of bulk/fluid manufacturing systems can be accomplished by developing specialized methods which handle the flow of material in a discrete event manner. This methodology allows for a simulation which runs in a timely manner so that conclusions can be drawn and fiscally sound decisions about capital improvements, labor reorganizations, maintenance schedules, etc. can be supported with data. This type of specialized simulation



Row	SKU	Production
0	Soap A1	53348
1	Soap A2	73619
2	Soap A3	28571
3	Soap A4	28143
4	Soap B1	9524
5	Soap B2	9381
6	Soap B3	19048
7	Soap B4	19034
8	Soap B5	23793

Figure 3: Soap Manufacturing System Model Worksheet

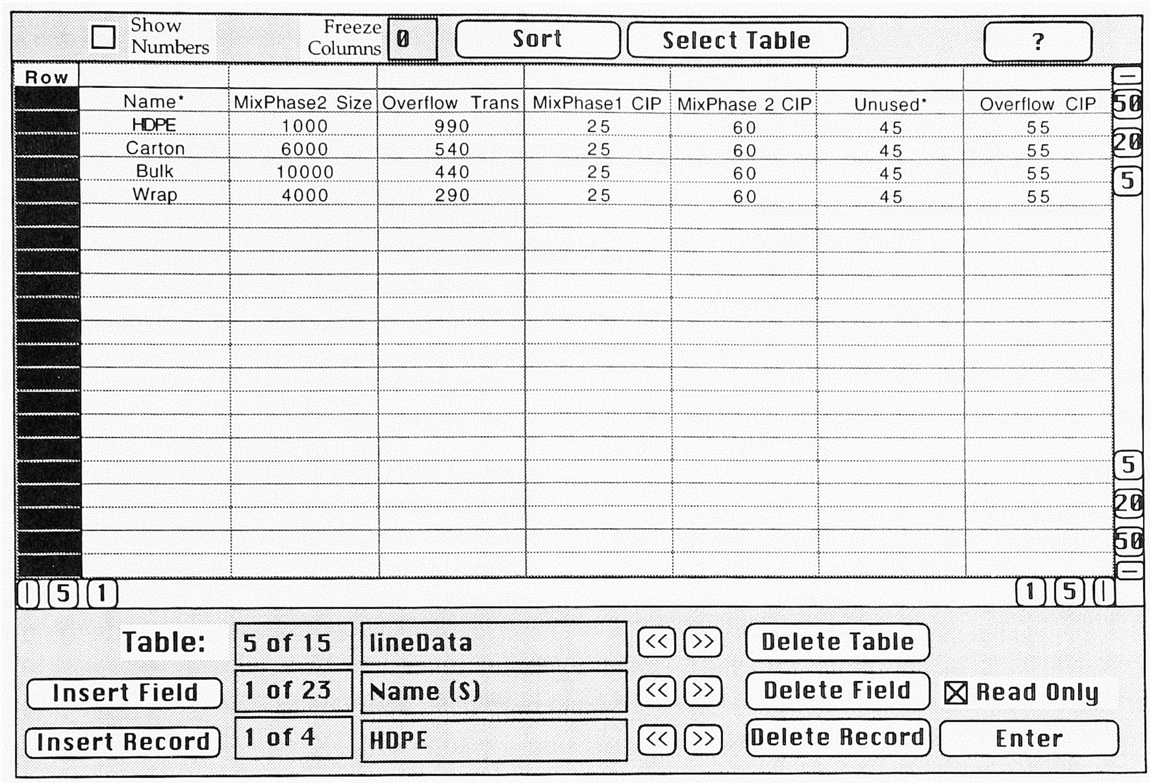


Figure 4: A Typical Database Table

architecture can be completely developed in the user-friendly environment of Extend, so that management-level individuals can run "what-if" scenarios with their facilities before making decisions.

REFERENCES

Diamond, B. 1993. Extend: A Library-Based, Hierarchical, Multi-Domain Modeling System. In *Proceedings of the 1993 Winter Simulation Conference,* ed. G. W. Evans, M. Mollaghasemi, E. C. Russell, and W. C. Biles, 240-248. IEEE, Piscataway, NJ.

Krahl, D. 1994. An Introduction to Extend. In *Proceedings of the 1994 Winter Simulation Conference,* ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila, 538-545. IEEE, Piscataway, NJ.

AUTHOR BIOGRAPHIES

ANDREW J. SIPRELLE is President and founder of Simulation Dynamics, Inc., a firm which provides simulation consulting, training, and custom models. Mr. Siprelle's industry experience includes creation and analysis of models for strategic and capacity planning, market analysis, and the application of industrial statistics. Applications of simulation have been in the areas of business, manufacturing, government and service operations. Before starting Simulation Dynamics he worked at the Aluminum Company of America. He received his B.S. in Industrial Engineering & Operations Research from Virginia Polytechnic Institute.

DAVID J. PARSONS' experience with simulation began in 1965 with experiments in the use of natural selection algorithms to evolve architectural designs. In 1981, he developed a simulation of seasonal heating system performance for the Wisconsin Gas Company. During the 1980's he designed, built and operated several dairy processing plants using simulation of key systems as an integral tool for design, value engineering and trouble shooting. Over the last four years he has applied Extend to continuous models for industrial clients and worked with faculty and dissertation students to model enzyme kinetics. Mr. Parsons received a B.A. from Harvard College and a Master of Architecture degree from the Harvard School of Design.