Modeling a Microbrewing Facility Using Discrete Event Simulation

Andrew J. Siprelle
Simulation Dynamics, Inc.

I. 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 a library based, iconic-block program (Extend) can be used to address such problems. A model of a microbrewing operation illustrates the use of this approach for determining the financial and operational impact of expansion projects requiring significant capital.

II. Introduction

The craft-brewed beer industry has enjoyed a rapidly growing popularity in recent years, as more consumers now favor finely crafted beers with the thought of quality instead of quantity. Businesses in this industry often start small operations with on-premise sales (brew pubs) to capture quick return. As these operations succeed, they quickly seek to add capability for kegging and bottling to capture markets for off-premise sales of their own brands and engage in contract brewing for other brewing facilities. To grow successfully, 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 flavors 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.

III. The Brewing System and the Use of the Model

Beer production starts in a two-section tank called the mash / lauter tun. Grain and hot water are mixed in the top section of the mash / lauter tun and heated during the conversion stage. The next step is lautering, which is the process of rinsing the grain with hot water stored in the bottom section of the mash / lauter tun. During lautering the liquid portion, called wort, is pumped to another tank, the kettle, where additional ingredients such as hops are added and the mixture is boiled. While lautering takes place, both the mash / lauter tun and the kettle / whirlpool are in use, and therefore, neither can be used for another batch at this time. The mixture is then clarified (solids are removed) by stirring it in a whirlpool-type motion. The wort is then pumped through a heat exchanger for cooling and on to one of several fermenting tanks, where yeast and perhaps other ingredients are added. There the liquid ferments for a specified period of days. Once fermentation is complete, the yeast is "blown down" from the tank, and the liquid remains in the fermenting tank for a number of days of aging. Following aging, the beer is pumped through a filter to one of several serving tanks. The beer is then fed from the serving tanks to a serving bar, kegging line, or bottling line.

Beers procede through each of the steps detailed above ,however, the production of different types and flavors of beer requires different processing times in the mash / lauter tun, the kettle / whirlpool, and the fermentation tanks. For example, ales only require 5 days of fermentation, while lagers require about 10 days. Ales also require only 5 days of aging before being served, while lagers require as much as 14 days. The particular system modeled in this brewpub transfers about 18 barrels of wort to the kettle for boiling and then produces about 14.5 barrels (after evaporation) into a serving tank.

A schedule for these steps must be designed to satisfy demand, provide for labor, and to make sure both materials and equipment such as tanks and flow lines will be available for different flavors of beer to be processed. Besides needing to ensure that different beers 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.

The puzzle becomes more complex as a brewing operation expands from an on-premise consumption system (a brew pub) to an operation which supplies kegs or bottles for off-premise consumption. Some of the crucial questions that need answering are as follows:

- Can the brewing equipment keep up with demand for both off-premise and on-premise sales?
- Is it wise to spend capital for kegging or bottling?
- How should labor be scheduled for brewing, kegging, and bottling?
- What is the best sequence for producing the various flavors?
- How much extra storage capacity is required for kegging and bottling?

To faithfully characterize the brewing 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.
- Adding or deleting ingredients.

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

IV. 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.

V. Extend and the Bulk Flow Architecture

Extend is a graphical, interactive, general purpose simulation program developed by Imagine That!. 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 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.

VI. Building the Model

The model of the brewpub facility 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.

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. Also, because of the object-oriented nature of the blocks, a user can easily add additional capacity in the form of another fermenting tank or serving tank, etc., and experiment with the expected operation of the facility.

A database developed within Extend is used to relate beer product to key process data. Providing access to a centralized database helps to tie blocks together without the need for excessive connections. A spreadsheet may be used to prepare the data, or data can be changed directly in the model using blocks which present views of database tables. A central "Control Panel" is used to access database data and run experiments.

VII. Results of the Simulation

Using the simulation model has shown that installation of new kegging and bottling equipment, and expectation of increased sales, requires more flexible storage requirements than previously thought. The times required in fermentation and aging are so much larger than the length of time necessary for the actual brewing process (time in mash / lauter tun and kettle / whirlpool) that more fermenting and serving tanks are necessary in order to continue serving a variety of beer.

The results of the utilization trace above indicate that the installation of either a kegging or bottling line would over-run the fermenting and serving tanks unless additional capacity were added in those areas. Because measures of cost and revenue are related to process changes simulated by the model, a dynamic view of financial performance of alternatives is available. While the installation of additional equipment requires significant capital, management is more confident about the return on investment after evaluating financial and operational results of several simulated scenarios.

VIII. 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 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 critical 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.

Biography

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. He has used Extend to help numerous companies analyze operations and increase productivity. 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.