

ARGONNE NATIONAL LABORATORY

9700 South Cass Avenue

Argonne, IL 60439

ANL/MCS-TM-276

Process Modeling for High-Throughput Biology

by

E. D. Frank and R. L. Stevens

Mathematics and Computer Science Division

Technical Memorandum No. 276

October 2004

This work was supported by the U.S. Department of Energy under Contract W-31-109-ENG-38.

Argonne National Laboratory, a U.S. Department of Energy Office of Science laboratory, is operated by The University of Chicago under contract W-31-109-Eng-38.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor The University of Chicago, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or The University of Chicago.

Available electronically at <http://www.osti.gov/bridge/>

Available for a processing fee to U.S. Department of Energy and its contractors, in paper, from:

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
phone: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

CONTENTS

ABSTRACT	1
1 INTRODUCTION	1
2 CASE STUDIES	1
2.1 MCSG CASE STUDY	1
2.2 BRADBURY LABORATORY CASE STUDY.....	3
3 APPLICATIONS OF PROCESS MODELING TO HIGH-THROUGHPUT BIOLOGY	4
3.1 PROCESS ABSTRACTION AND ANALYSIS	6
3.2 DERIVATION OF IMPLEMENTATION	6
3.3 DERIVATION OF INFORMATICS NEEDS.....	7
3.4 DERIVATION OF SCHEDULING AND ALARMS	7
3.5 SIMULATION AND OPTIMIZATION	8
ACKNOWLEDGMENTS.....	8
REFERENCES	8

Process Modeling for High-throughput Biology

E. D. Frank and R. L. Stevens

Abstract

We interviewed researchers at two high-throughput biology laboratories in order to identify and to understand their workflows. We constructed and studied a detailed process model for one of the labs. Based on these efforts and on the benefits of the model to the experimenters, we discuss here ideas for applying process modeling to high-throughput biology laboratories.

1 Introduction

Process models are models of the activities that go on in a system. The system might be a business, laboratory, program, or machine. “Process” refers to activities in the system. We use the term “models” because they do not necessarily have every detail of the actual system and, indeed, we may discard or simplify aspects that are not relevant to the questions being addressed. For example, a model of customer queues at a bank might ignore all the details of operations at a teller window (such as greetings, identification checking, and form completion) and replace these operations with a single delay-time distribution if queue depth is the major question; however, these details might be added if one was predicting the delay distribution or if those substeps could be interrupted.

We have studied two high-throughput biology laboratories to investigate how process modeling can be applied. Section 2 summarizes the individual workflows. In brief, the high throughput procedures in the case studies involved batched processing of samples (e.g., in 96-well microtiter plate format) and involved numerous movements of samples from one vessel type to another. The fundamental protocol was in terms of what was done to a single sample: The details about plate handling and sample tracking were related more to the equipment on the floor than to the chemistry or biology. In Section 3, we describe several applications of these case studies, ranging from process abstraction to optimization.

2 Case Studies

We studied the cloning and expression pipeline from the Midwest Center for Structural Genomics (MCSG) at Argonne [1, 2] and studied the Bradbury Laboratory pipeline for high-throughput production of phages for phage-display-based affinity tags [3].

2.1 MCSG Case Study

The MCSG cloning and expression pipeline was presented to us in detail, listing most operations done on each sample. We used a commercial process-modeling tool, SimProcess [5], to model this pipeline.

The first question we posed was whether we could reproduce the facility’s throughput and the sample transit time with the model. For this question, much of the detail of the system was irrelevant. For example, a series of steps in the laboratory description—Grow, Induce, Lyse, Centrifuge lysis plate—could be replaced with the single step Induce. This approach improves precision because the timing of aggregates is better known than the timing of shorter substeps and the uncertainties of these accumulate.

Indeed, there is no reason not to roll up all the details into a few top-level boxes. Of course,

whenever decision making, like quality control, can cause iterations on a sample, one must insert a breakpoint in the flow. All steps between these breakpoints can be rolled up for the purposes of computing transit time.

Figure 1 shows the high-level processes identified when this approach was adopted. These processes—DNA production, cloning, Immunoassay, and screening—are refined to add essential details. For example, Figure 2 shows additional details for the Immunoassay box. Here, it is possible to see where details have been rolled-up. In particular, the icon “Δt” labeled “Grow, Induce, Centrifuge” combines four steps from the full process description.

When we approached the model from the viewpoint of breakpoints in the flow corresponding to decision making, we detected that the system lacked control protocols at the highest level. For example, under what conditions are new samples entered into the system? In various places where samples queue, what policies determine which entity moves out of the queue first? Are there fast-track scenarios where the facility stalls while a rush order moves through? Further interviews determined that these aspects of the facility are handled informally. Decisions are made according to available resources and current objectives. Thus the modeling process revealed undocumented protocols and made clear where automation was lacking. The MCSG staff found this information extremely valuable.

The second question we posed in the case study concerned resource contention in the facility: What resources, such as people and robots, determined throughput? Was the process scalable?

Here again, we were able to discard details unrelated to transitions from resource to resource; steps between resource transitions can be rolled up. We followed several resources: a MultiMek robot, a BioMek robot, and lab personnel. The model tracked transitions from resource to resource and resource down times (such as weekends and evenings) and had as input the numbers of each resource and the transit times between the steps. Resource utilization percentages and contention also were tracked.

The first version of our model predicted that most resources were largely idle, including staff, a prediction that was certainly not true. Further interviews indicated that pieces of the protocol were not in the original diagram and were done opportunistically by lab personnel. Data entry, sample labeling, and reagent preparation are examples. Because these are done opportunistically, they cannot be automated, by definition. They represent disconnects in the overall system flow. This approach does maximize the individual efficiency of the staff personnel, but the model revealed that the manual operations were rate-limiting the system. As a result, the robots could not be employed at high utilization.

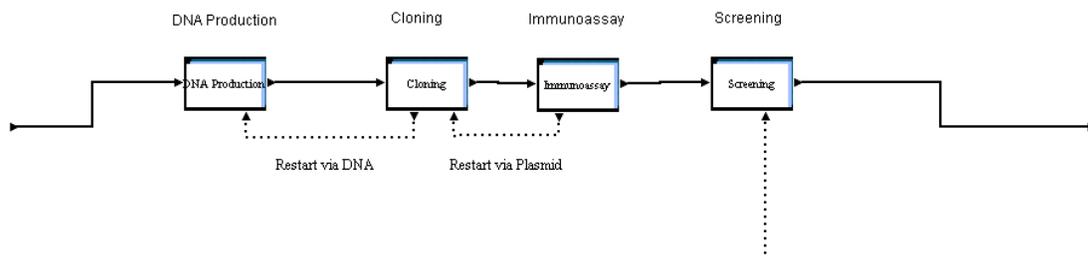


Figure 1. Top level of MCSG model

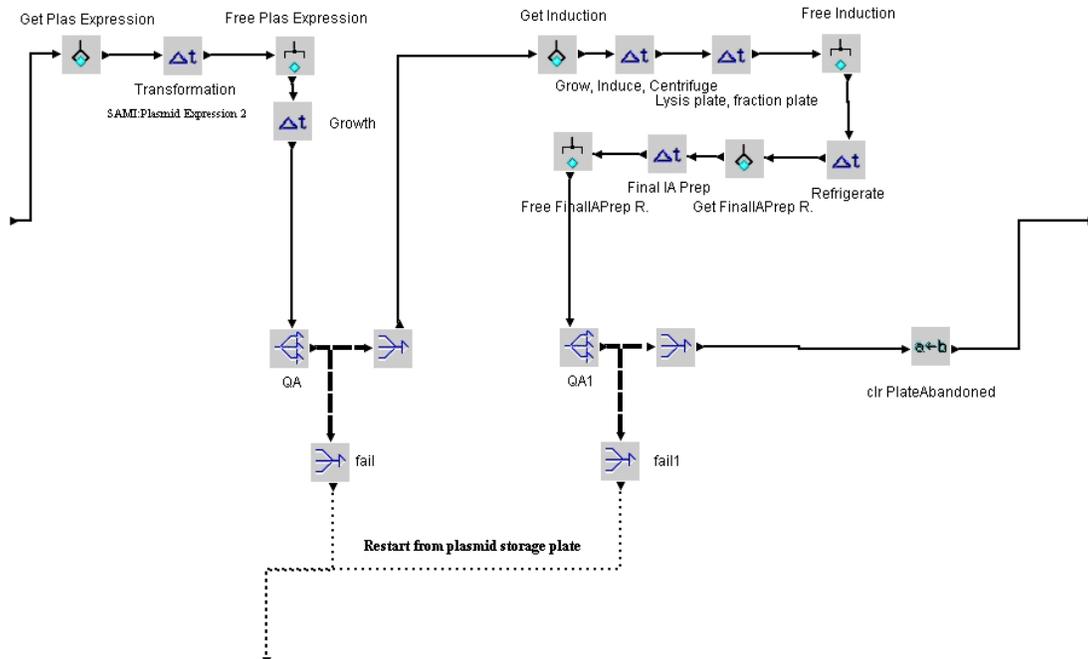


Figure 2. Example of defining the details corresponding to a box in the top-level description from Figure 1. The details for the box “Immunoassay” are shown.

A better approach would be to follow a fixed process flow, even if that reduces the efficiency of an element of the system. The reason is that replacing the opportunistic approach with a constant process allows automation and allows either the replacement of people with machinery or the easy incorporation of additional people.

In summary, this case study revealed that the level of detail needed to capture the chemistry protocols is greater than that needed to test for transit times and resource utilization. It also showed that the formulation of the model depends on the questions being asked. Intuitively, one feels that the transit-time and resource models should be derivable from the chemistry processes. The model also underscored the fact that an informatics system to acquire data and print labels would help remove bottlenecks.

2.2 Bradbury Laboratory Case Study

Figure 3 shows a high-level diagram of the phage-display production pipeline at Bradbury Laboratory. This case study adds two new factors relative to the MSCG case study. First, there is a loop in the processing that is fundamental to the protocol: loop over amplification and filtering until a positive ELISA is obtained. Second, since the process is still under design, there are alternative implementations of various steps, for example, selection via binding to Nunc TSP plastic bins vs. biotinylation to magnetic beads and variations on monoclonal assay technique such as ELISA vs. filtering.

The Bradbury Lab also involves additional vessel types beyond the MSCG's list of 96-well plates, 96-well culture plates, and culture tubes. In addition, there are the Nunc TSPs, omni-plates, Q-trays, and new transformations beyond transfers, aliquoting, and so forth, in particular titration.

The diagram in Figure 3 was provided by the experimenters. It clearly is not at a level of detail to program robots or track samples, but it does accurately convey the essential logic of the process. It omits details such as tracking well-to-well relationships between plates. Such a diagram is a

natural way for experimenters to plan. At this level, the focus is on per sample flows and many aspects of the biochemistry such as elution, washing, and induction are ignored

3 Applications of Process Modeling to High-Throughput Biology

The case studies suggest how to apply process modeling to high-throughput biology.

- Process abstraction and analysis
- Derivation of implementation
- Derivation of informatics needs
- Derivation of scheduling and alarms
- Simulation and optimization

We explore these in the following five subsections.

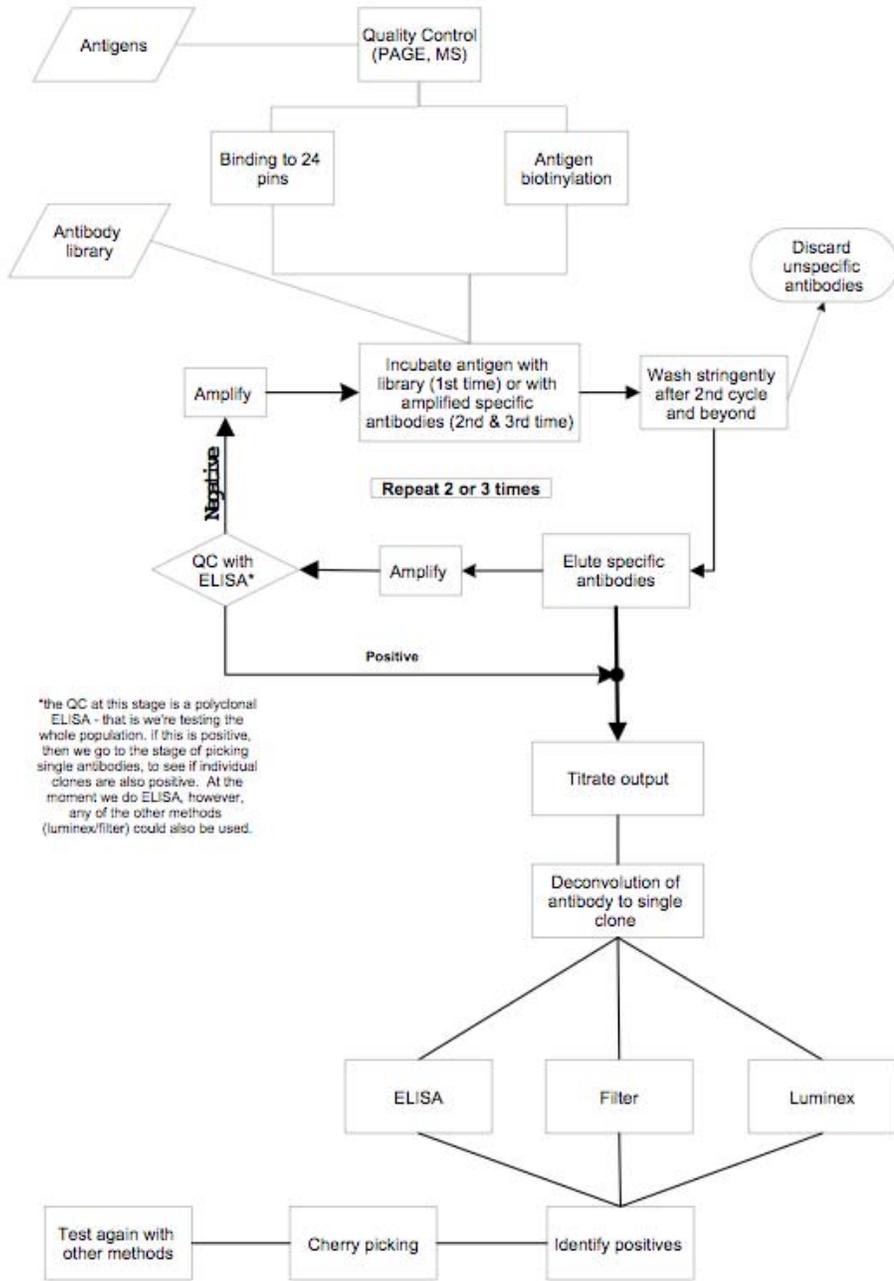


Figure 3. Process Flow for the Bradbury Lab

3.1 Process Abstraction and Analysis

Actual high-throughput processes handle samples in parallel (e.g., many samples on a 96-well plate). *Process abstraction* focuses on per sample descriptions of the protocol. It provides a designer with a design language for describing basic steps such as elution, washing, and incubation, as well as testing and branching the flow based on the test values.

The language also allows cross-references between steps in the processing of a sample. For example, consider a sample that has been aliquoted into N test samples that are screened against a library of some kind. The subsequent assay may take a number of steps, yielding a final measurement on which a pass/fail decision is made for the corresponding member of the library. The design language allows the designer to express a cherry-picking operation that relates a positive test result back to an earlier step in the protocol so that the corresponding sample can be resampled and carried forward for further processing.

The designer will not worry about batching of samples onto 96-well plates or other media. The designer will specify types of apparatus, like incubator or SDS-PAGE, plus requirements for that instrument, but not specify a specific instrument.

A library of protocols would be available to the process designer that could be inserted as design blocks. This would allow substeps among independently designed processes to be handled systematically and would enable reuse of pieces of the design. Examples might include titration, SDS-PAGE, heat-shock uptake, or specific kinds of induction. In this way, the abstract design could have multiple levels of detail.

Subsequent sections discuss using this process model to derive implementation, scheduling, and database needs and to perform simulation.

3.2 Derivation of Implementation

Given a process model, as just described, and given a description of available equipment in the facility, one can derive an implementation of the abstract protocol for a facility.

First, the abstract protocol is analyzed to determine the kinds of apparatus needed and their individual requirements. The facility description is queried for a list of its installed equipment with descriptions of their capabilities, capacities, and acceptable media. This list is matched to the abstract protocols requirements list, giving a set of candidate equipment.

Descriptors of the equipment enumerate allowable media for each, for example, 96-well plates on some equipment and N-fold racks of individual tubes on other equipment. From this information, a batching schedule can be computed whereby multiple samples are put onto single 96-well plates or other high-throughput media.

It is necessary to transform between media types in order to bring samples from one apparatus to another. For example, if a robot is used for preparatory work on 96-well plates leading up to SDS-PAGE, but only 16 samples can be put on a gel, then a step is needed to transform the 96-sample batching into 16-sample batching. Or, samples may be in a 96-well plate and need to be grown out, thus requiring a transfer into a 96-well deep-well plate.

The implementation derivation proceeds through these computations until all abstract resources from the abstract protocol design are mapped to kinds of apparatus in the facility and continues until all transformations omitted in the abstract protocol design have been added as needed to achieve a complete, end-to-end processing pipeline.

The derived implementation may not be unique. Heuristics and optimizations are needed to make choices.

Some equipment (e.g., robots) is programmable but often through proprietary software that

perhaps cannot be used by the implementation generator. In this case, there will be a library of preinstalled programs for the robot and descriptors for each of the programs. These programs are one to one with the design library elements. In a sense, the robot is replaced by a set of transformations it can achieve, and these are used as the building blocks by the implementation generator. Even in the case of arbitrarily programmable robots, this may be a preferable approach to limit the amount of variability in implementation, thus increasing the uniformity of processing done in the facility and increasing the value of production monitoring.

3.3 Derivation of Informatics Needs

Informatics tracks the processing of samples in the facility. Completed samples should have detailed logs of how they were processed, and staff should be able to query the status of samples in progress.

The implementation derived from a process model is a complete enumeration of the steps that will be carried out, the kinds of equipment that will be used and, tacitly, the data that will be generated. This enumeration can be analyzed to determine the corresponding informatics steps that are needed.

For example, the processing of a sample is a series of steps each of which involves some kind of sample on some kind of media on some kind of apparatus. A unique identifier can be assigned to each of these; the processing history becomes a list of those identifiers. Relations in the database that locate that list given a sample id and relations that map those identifiers to media instance records (a 96-well plate bar code, for example) and sample locators on those media (a well address, for example) could represent the processing. Relations can tie those identifiers to generated data that has been loaded into the database as a result of the processing steps.

The intention here is not to propose an informatics schema but to indicate the close relationship between the implementation description and the informatics needs. The set of design primitives in the abstract process language and the set of instrument descriptions for the facility combine to specify the data space that must be accommodated by any proposed informatics schema. Also, if only generated implementations are run, then an informatics system will handle the resulting tracking.

3.4 Derivation of Scheduling and Alarms

Consider a facility in operation. Some equipment is in use, some free. If a set of samples is to be processed, time must be scheduled on the equipment. Such scheduling should be done to optimize throughput or sample quality (batching of-a-kind processing might, for example, reduce concerns about variability in reagents).

The implementation derived from the process model is in terms of kinds of instruments. Scheduling assigns specific instances of those instruments in the facility to the job and at certain times. The processing pipeline in the derived implementation can thus predict what equipment is needed when.

Process models created from library (sub)protocols offer an advantage for schedule prediction. If library modules are reused, and since sample entry and exit from the library module's corresponding pipeline is clearly defined, the library provides a convenient definition for accumulating statistics for processing time. This can be used to help predict how long pipelines constructed from those elements will take to run. It can also help predict when the associated equipment will become busy and free through the course of the run.

Alarms can be derived from this information. For example, a process model's derived implementation explicitly states what steps will occur. This information can be used to audit sample histories and to generate alarms for missing information. If processing times are predicted, stalled samples can be identified.

We acknowledge that it is not clear whether processing times can be predicted. Further research is required. If loops exist in the processing, as in the Bradbury Lab's protocol, execution of the protocol can generate new samples. Combine this situation with steering based on measurements and interleaving of jobs in the facility, and concerns about chaotic transit times arise. The "Beer Game" is a classic supply-chain optimization game that exhibits chaotic behavior.

3.5 Simulation and Optimization

Simulations of implementations derived from process models can address several issues.

If the connection between informatics, process models, and derived implementation is achieved as described, a simulation can generate a sequence of database operations and times. These can be connected to an implementation of the informatics system to serve as test vectors. Execution of these will allow a test database to accumulate complex state that will be a more stringent test of the system than can be achieved with manual unit tests. It is not clear, however, whether concurrency tests resulting from these test vectors would be believable.

Simulations of process models might help one understand sample transit time dispersion, especially if it proves possible to simulate multiple processes running at once to gauge their interactions. Transit time studies will require calibration data for transit times across the various steps in the process model, and those data may not be reliable for some time. Attempting the simulations before hand could help determine which data is needed from monitoring.

Related to transit times, simulations can track when resources like robots are busy and free and compute percent utilization, queue depths, wait times, and so forth.

Acknowledgments

This work has benefited from the efforts of Andrew Bradbury, Frank Collart, Lynda Dieckman, Denise Holze, and Andrzej Joachimiak.

References

- [1] L. Stols, M. Gu, L. Dieckman, R. Raffin, F. R. Collart, and M. I. Donnelly, "A new vector for high-throughput, ligation-independent cloning encoding a tobacco etch virus protease cleavage site," *Protein Expr Purif*, vol. 25, pp. 8-15, 2002.
- [2] L. Dieckman, M. Gu, L. Stols, M. I. Donnelly, and F. R. Collart, "High throughput methods for gene cloning and expression," *Protein Expr Purif*, vol. 25, pp. 1-7, 2002.
- [3] A. R. Bradbury, Los Alamos National Laboratory. Private communication, 2004
- [4] A. R. Bradbury and J. D. Marks, "Antibodies from phage antibody libraries," *J Immunol Methods*, vol. 290, pp. 29-49, 2004.
- [5] SimProcess, CACI International, Inc., 2004