

**DISCRETE-EVENT SIMULATION OF A
PULL-DRIVEN MATERIALS-HANDLING PROCESS
THAT REQUIRES RESOURCE MATCHING:
EXAMPLE OF PIPE-SPOOL INSTALLATION**

by Iris D. Tommelein

Technical Report 97-2

March 1997

**Construction Engineering and Management Program
Civil and Environmental Engineering Department
University of California, Berkeley, CA 94720-1712**

DISCRETE-EVENT SIMULATION OF A PULL-DRIVEN MATERIALS-HANDLING PROCESS THAT REQUIRES RESOURCE MATCHING: EXAMPLE OF PIPE-SPOOL INSTALLATION

by Iris D. Tommelein

ABSTRACT

Many construction processes require installation of unique materials in specific locations in the facility being built: materials and locations must match before installation can take place. Mismatches due to delay and uncertainty in supplying materials or completing prerequisite work at those locations hamper field productivity. To illustrate this, a model is presented of a materials-handling process with a matching problem that is typical in fast-track process-plant construction projects, namely hanging pipe spools in work areas. Because of the uniqueness of materials and locations combined with the unpredictability in duration and variation in execution quality of various steps in the supply and installation process, there are different ways to sequence and control the delivery of spools and the completion of work areas. This report describes alternative ways for defining such sequencing and control. The impact these alternatives have on process execution is illustrated by means of deterministic and probabilistic computer-based process models. One probabilistic model reflects total lack of coordination between spool delivery and work area completion prior to the start of construction; a second probabilistic model describes perfect coordination. The consequences in terms of materials buffer sizes and construction progress are reflected in the outcome of discrete-event simulation of those models. A third probabilistic model then illustrates the use of a technique derived from lean production, called pull-driven scheduling. It involves real-time feedback regarding progress status on site to materials suppliers off site, which is used to opportunistically re-sequence process steps. The outcomes are reduced buffer sizes and more timely project completion and, when properly accounted for, increased productivity.

© Iris D. Tommelein, March 1997.

TABLE OF CONTENTS

1. INTRODUCTION	1
2. RELATED WORK IN LEAN CONSTRUCTION	2
3. PUSH-DRIVEN VS. PULL-DRIVEN PROCESS MANAGEMENT	4
3.1. PUSH-DRIVEN PROCESS MANAGEMENT	4
3.2. PULL-DRIVEN PROCESS MANAGEMENT	5
4. EXAMPLE PROCESS SCENARIO: INSTALLING PIPE SPOOLS	6
5. INDUSTRY PRACTICE	6
6. PROCESS MODELING	8
6.1. PROCESS-MODEL REPRESENTATION	8
6.2. PIPE-SPOOL PROCESS MODEL	10
6.2.1. <i>Off-Site Activity Sequence</i>	12
6.2.2. <i>On-Site Activity Sequence</i>	14
6.2.3. <i>Joint Activity Sequence</i>	15
7. IMPLEMENTATION AND SIMULATION OF ALTERNATIVE MODELS	16
7.1. DETERMINISTIC MODEL	16
7.1.1. <i>Model Parameters</i>	16
7.1.2. <i>Pipespool Buffer Size</i>	18
7.1.3. <i>Productivity of Installation Crew</i>	18
7.1.4. <i>Project Duration</i>	20
7.2. ALTERNATIVE PROBABILISTIC MODELS	20
7.2.1. <i>Model Construction</i>	20
7.2.2. <i>Data Collection and Representation</i>	20
7.2.3. <i>Model A - Random Sequencing</i>	21
7.2.3.1. <i>Model Parameters</i>	21
7.2.3.2. <i>Pipespool Buffer Size</i>	21
7.2.3.3. <i>Productivity of Installation Crew</i>	21
7.2.3.4. <i>Project Duration</i>	22
7.2.4. <i>Model B - Coordinated Sequencing</i>	24
7.2.4.1. <i>Model Parameters</i>	24
7.2.4.2. <i>Pipespool Buffer Size</i>	24
7.2.4.3. <i>Productivity of Installation Crew</i>	25
7.2.4.4. <i>Project Duration</i>	25
7.2.5. <i>Model C - Pull-driven Sequencing</i>	27
7.2.5.1. <i>Model Parameters</i>	27
7.2.5.2. <i>Pipespool Buffer Size</i>	28
7.2.5.3. <i>Productivity of Installation Crew</i>	28
7.2.5.4. <i>Project Duration</i>	28
8. IMPLEMENTATION HARDWARE AND SOFTWARE	30
9. CONCLUSIONS	30
10. ACKNOWLEDGMENTS	31
11. APPENDIX 1: REFERENCES	31
12. APPENDIX 2: SOURCE CODE FOR PIPE-SPOOL MODEL	34

LIST OF FIGURES

FIGURE 1: % PLANNED VS. ACTUAL IN PERCENT OF TOTAL OF ISOMETRIC DRAWINGS (ISOS), FABRICATED SPOOLS DELIVERED TO SITE, AND SPOOLS IN PLACE	7
FIGURE 2: STROBOSCOPE PROCESS MODEL OF PIPE-SPOOL INSTALLATION	11
FIGURE 3: SINGLE RUN PERCENT COMPLETE VS. TIME FOR COORDINATED SEQUENCING IN DETERMINISTIC MODEL	17
FIGURE 4: SINGLE RUN NUMBER RESOURCES VS. TIME FOR COORDINATED SEQUENCING IN DETERMINISTIC MODEL	18
FIGURE 5: 1000 ITERATIONS PERCENT STAGEDSPOOL VS. TIME FOR ALL PROBABILISTIC MODELS ..	20
FIGURE 6: SINGLE RUN PERCENT COMPLETE VS. TIME FOR RANDOM SEQUENCING	22
FIGURE 7: SINGLE RUN NUMBER RESOURCES VS. TIME FOR RANDOM SEQUENCING	23
FIGURE 8: 1000 ITERATIONS NUMBER STAGEDSPOOL VS. TIME FOR RANDOM SEQUENCING	23
FIGURE 9: 1000 ITERATIONS PERCENT AREADONE VS. TIME FOR RANDOM SEQUENCING	24
FIGURE 10: SINGLE RUN PERCENT COMPLETE VS. TIME FOR COORDINATED SEQUENCING	25
FIGURE 11: SINGLE RUN NUMBER RESOURCES VS. TIME FOR COORDINATED SEQUENCING	26
FIGURE 12: 1000 ITERATIONS NUMBER STAGEDSPOOL VS. TIME FOR COORDINATED SEQUENCING ..	26
FIGURE 13: 1000 ITERATIONS PERCENT AREADONE VS. TIME FOR COORDINATED SEQUENCING	27
FIGURE 14: SINGLE RUN PERCENT COMPLETE VS. TIME FOR PULL-DRIVEN SEQUENCING	28
FIGURE 15: SINGLE RUN NUMBER RESOURCES VS. TIME FOR PULL-DRIVEN SEQUENCING	29
FIGURE 16: 1000 ITERATIONS NUMBER STAGEDSPOOL VS. TIME FOR PULL-DRIVEN SEQUENCING	29
FIGURE 17: 1000 ITERATIONS PERCENT AREADONE VS. TIME FOR PULL-DRIVEN SEQUENCING	30

LIST OF TABLES

TABLE 1: SELECTED STROBOSCOPE SYMBOLS	9
TABLE 2: DETERMINISTIC VS. PROBABILISTIC MODELS	17
TABLE 3: ALTERNATIVE SEQUENCING STRATEGIES	20

1. INTRODUCTION

Construction involves handling materials and installing them according to project specifications and design drawings in the facility being built. Data describing the flow of materials in the supply chain, that is, describing when and where materials are being engineered, fabricated, transported, and staged on site is essential in planning and executing installation work. Tracking materials through the supply chain requires more or less detail depending on whether the material of concern will be available in large quantities of identical units (e.g., concrete blocks; drywall; electrical conduit; nails, nuts, and bolts) or small quantities of units with unique properties (e.g., an air conditioning unit, a main entrance door, or a pipe spool). Along this spectrum lie materials supplied in modest quantities, possibly with some degree of interchangeability (e.g., windows, exterior vs. interior doors, structural steel, timber in precut lengths). Availability characterizes not only physical resources such as materials or equipment to be installed, but also space on site, for instance work areas where installation will take place.

Field installation crews, responsible for the final step in the materials flow process, must find resources that match among those available to them; they must ensure that the right material gets put in the right place. For instance, they must select a specific material (e.g., pipe spool SP-123), retrieve the correct installation accessories (e.g., various attachments and supports), and match them in accordance with the location where assembly is to take place (e.g., area AR-123). An integral part of their work, time and again, is to solve the so-called "matching problem." Because facilities comprise many thousands—if not tens- or hundreds of thousands—of materials, tackling the matching problem is an enormous task, but those executing the last step in the process, namely installation, cannot avoid addressing it.

In contrast, those responsible for engineering and design, fabrication, supply, and site storage of materials, as well as construction managers overseeing the project often overlook the matching problem that installation crews face. Dealing with materials on an item-by-item basis means paying attention to minute details. It is a tedious task, largely irrelevant to their own. Accordingly, selected matching-problem details are abstracted away by each party so that they can focus on problems of more direct concern to them. For example, structural designers do not worry about vendors' ability to deliver specialty valves or nuts-and-bolts because it is outside of their scope of work. Pipe-spool fabricators optimize production schedules to suit their plant's fabrication constraints and to all projects they have going on concurrently. Shipping agents pack materials on trucks so that loads will be stable and meet weight and dimensional limits encountered during transportation; they optimize travel given load- and truck-fleet sizes while meeting delivery schedules. Laydown yard personnel group materials by shipment (e.g., Tuesday's deliveries were put in area L-1) or type (e.g., all valves are kept in area M-4) which eases tracking. Only in exceptional situations do they organize them by their final-installation destination (e.g., small bulk items may get bagged by warehouse personnel before being brought to the work face). Project management controls progress based on percentages-of-total of materials engineered, delivered to the site, or installed.

Because of this abstraction, installation crews rarely have the data they need to

optimally schedule and thus execute their work. They must rely on assumptions embedded in schedules made prior to the start of construction. How much of a problem this creates depends on the extent to which uncertainties in the supply of what they need manifest themselves during project execution. If project schedules were well thought-out and all steps prior to installation had no uncertainty in duration or execution quality associated with them, then matching would be easy. In practice, unfortunately, this is not the case. Many projects are executed on the fast track, so construction starts long before design has been completed or materials deliveries have been sequenced. Installation crews and equipment are often kept waiting because delays in materials supply and delays in completing prerequisite site work lead to mismatches that foul up scheduled work sequences (e.g., steel for a building's fifth floor may be on site but that for the first floor, still to be erected and supporting higher levels, is not). This lowers the installation crew's productivity and extends the project's duration of construction.

In order to increase understanding of these issues, a model was created of a process that is characteristic of the process-plant sector of the construction industry. Alternative strategies for sequencing materials deliveries are presented in this report and their execution was simulated so computer data supports the comparison between them.

2. RELATED WORK IN LEAN CONSTRUCTION

Matching problems pose unique requirements on a production system. When parts to be assembled are all one of a kind, unique identifiers will be required and parts will have to be tracked on an individual basis through assembly. These requirements contrast sharply with those provided by mass production. In mass production, production managers optimize the use of machines requiring large setup times by running them for a long time, producing huge numbers of identical parts, before changing over to a new setup. This results in a large build-up of in-progress inventories in quantities to meet projected and therefore uncertain demand, and to be maintained at a cost that supposedly is offset by machine efficiency. In contrast, lean production advocates one-piece flow to get the right materials to the right place and in the needed quantity, without the potential for creating such potential excesses.

The lean production philosophy focuses on adding value to a raw material as it proceeds through various processing steps to end up as a finished product. Specifically, it advocates the avoidance, elimination, or at least reduction of "waste" from this so-called value stream. Ohno, who first articulated this philosophy and implemented it in the Toyota production system, classified sources of waste as follows (8 added by Womack and Jones 1996): (1) Defects in products; (2) Overproduction of goods not needed; (3) Inventories of goods awaiting further processing or consumption; (4) Unnecessary processing; (5) Unnecessary movement of people; (6) Unnecessary transport of goods; (7) Waiting by employees for process equipment to finish its work or for an upstream activity to complete; and (8) Design of goods and services that fail to meet user's needs.

The lean production philosophy, since it emerged in the 1950s, has had a major impact in the Japanese manufacturing industry. Its benefits gradually became convincing outside of Japan as well. Since the 1980s, manufacturing companies in the

US have been adopting this philosophy. They converted their operations to implement lean production techniques and, consequently, were able to radically increase their competitiveness (Womack and Jones 1996). Some lean production techniques are:

1. Stopping the assembly line to immediately repair quality defects. While this usually is very disruptive for the process as a whole, there are several advantages to doing so: (1) the flawed processing step can be corrected right away, before numerous other assemblies have undergone the same treatment; (2) it is easier and less costly to discover and repair a quality defect early on in a process rather than at the end, after an assembly has been completed.
2. Pulling materials through the production system to meet specific customer demands, as opposed to building up inventories of work-in-progress and completed assemblies in anticipation of (often mis-judged) customer demand.
3. Reducing the overall process cycle time by minimizing each machine's change-over time, so that the economic benefit of mass production over single-piece production is reduced substantially, in turn making custom small-size batches easier to justify.
4. Synchronizing and physically aligning all steps in the production process so there is little wait time for people or machines, and virtually no staging of partially-completed products.
5. Clearly documenting, updating, and constantly reporting the status of all process flows to all involved, so each person knows what others do and understands the implications of quality of their own work on the quality of the process output.

Fairly recently, lean production has become a subject of interest in the construction industry (Koskela 1992). Undoubtedly, there is much waste in construction. As part of this effort to identify and minimize wasteful activity, this report focuses on the build-up of piles of materials on site, awaiting installation but of no immediate use because "matching parts" are missing. These in-progress inventories are created because site deliveries cannot be timed precisely, and, equally important, because the need for materials often remains ill-timed due to schedule uncertainties. Such piles not only occupy large amounts of space, but, also, a fair amount of effort goes into keeping track of what is available and where and capital is tied up to carry inventory.

Since the publication of Koskela's (1992) seminal report, researchers around the world have been studying lean production and its applicability to construction. The International Group for Lean Construction has organized annual conferences since 1993 (Alarcon 1997). Unfortunately, the translation of lean concepts from manufacturing to construction is not automatic because of the unique characteristics of the architectural-engineering-construction (AEC) industry in addition to the geographic diversity among projects.

As part of this effort, researchers in construction have begun to realize that construction management must include production control systems (Ballard and Howell 1997, Bernold and Salim 1993, Melles and Wamelink 1993) to complement those more high-level project management systems currently in use. Control systems must include not only activities being performed at the project site but also those that make

up the entire resource supply chain (O'Brien 1995). The work described in this report belongs to this school of thought.

Several researchers have been assessing the applicability of lean construction to construction. Howell et al. (1993) discussed how buffers of materials can alleviate the dependencies and worker idle time otherwise incurred when process sub-cycles interact with one another. Ballard and Howell (1997) implemented the "Last Planner" production system on several industrial sites (Howell and Ballard 1995) to shield installation crews from uncertainties in work flow. Phair et al. (1997) recently reported how equipment manufacturers are reducing set-up time by changing product designs (e.g., buckets and other attachments). Tommelein (1997a, b) showed how the pull technique with feedback regarding progress of on-site work to fabricators can improve the pipe installation process on oil-refinery construction sites. The latter work is detailed in this report.

3. PUSH-DRIVEN VS. PULL-DRIVEN PROCESS MANAGEMENT

3.1. Push-Driven Process Management

The traditional approach for planning construction work is to articulate activities and dependencies between them, and then to assign durations and resources to each activity. A schedule is then developed by calculating early and late activity starts and finishes using the Critical Path Method (CPM). Resource leveling or allocation algorithms may yield some adjustments to the early-start schedule, but upon project execution, activities are expected to start at their earliest possible date in order not to delay succeeding activities or the project as a whole.

Project controls aim at adhering to resulting schedule to the largest extent possible. This approach is based on the assumption that all resources required to perform an activity that is about to start will indeed be available at that activity's early-start time. In this so-called "push-driven" approach, each activity passively waits for its ingredients (instructions, labor, materials, equipment, and space) to become available, e.g., by being released by finishing, predecessor activities. When some have become available but others needed at the same time have not, those available will wait in a queue or buffer for the combination of resources—the set of "matching parts"—in its entirety to be ready. While it may be possible to start work with a incomplete set of resources, chances are this will negatively affect productivity (e.g., Thomas et al. 1989, Howell et al. 1993).

More often than not, however, because of uncertainty in duration as well as variation in execution quality and dependency logic of activities, schedule delays occur as construction work progresses. It may be possible to quantify this uncertainty during the planning stage. For instance, the Program Evaluation and Review Technique (PERT) takes as input probabilistic distributions to characterize activity durations. Yet, the actual manifestation of that uncertainty is known only upon plan execution and must thus be dealt with in real time. At that point, rigorously adhering to the initial schedule may not be the best approach for successful project completion as network

characteristics and resource availability will deviate from those assumed when that schedule was generated.

Moreover, traditional CPM schedules do not necessarily show individual resources and their allocation to activities. Certainly, procurement schedules highlight milestone delivery dates of major items, but most materials will arrive in shipments comprising several units whereas the schedule will reflect only groupings, without detailing individual parts. This makes it difficult for anyone to use the schedule as a guide for installation work that involves unique groupings of such parts. When missing parts are identified during the on-site allocation process, it is much too late to prevent work delays.

In addition, current expediting practice is to touch base on a regular basis with the engineering design firm, fabricator, supplier, or shipper of whom goods or services are expected. They are contacted prior to the deadline for completion of their work, in order to make sure the target delivery date, e.g., of key materials or pieces of equipment, will be met. Yet, formal procedures seldom are in place to expedite all that is needed or to systematically update the schedule and, if needed, reschedule activities when it can be anticipated that deadlines will not be met. Accordingly, the traditional, push-driven approach to scheduling construction work prior to project commencement with no corrective rescheduling as work progresses leads to process inefficiencies and a less-than-optimal project performance.

3.2. Pull-Driven Process Management

The main objective of a "pull-driven" approach to process management is to produce finished products as optimally as possible in terms of quality, time, and cost, so as to satisfy customer demand. Achieving high process throughput while minimizing operating expenses including in-process inventories is key. One should avoid working on a component, yet leave it unfinished for some time, not knowing if there will be any demand for it at all.

Keeping busy by processing just any one of the resources in the input queue of an activity requiring a combination of resources is insufficient. To pull means that resources must be selectively drawn from queues—so the activity that processes them will be busy just the same—but chosen so that the activity's output is a product needed further downstream in the process, and needed more so than its output using other resources in the queue would have been. Ideally, resources should wait in queues as little time as is possible.

To implement pull-driven approach, selective control is needed over which resources to draw for any given activity. This selection is driven by information not solely about resources in the queues immediately preceding the activity under consideration, but also about work-in-progress and resources downstream (successor queues and activities) in the process. In any one queue, resources will get priority over others in the same queue if they are known to match up with resources forecast to be or already available in queues further downstream in the process. This way, those available resources will not unduly await their match and be in process for any time longer than needed, though their planned processing sequence may be violated.

4. EXAMPLE PROCESS SCENARIO: INSTALLING PIPE SPOOLS

Consider the process of constructing an industrial process facility, such as an oil refinery, that involves installing many hundreds of unique pipe spools. This process is here characterized as comprising two chains of activities: pipe spools are designed and fabricated off-site while work areas are prepared on site. After spools have been shipped to the site, these chains merge upon the installation of spools in their designated areas.

Pipe spools are fabricated off-site according to the availability of engineering design information, the fabricator's plant production capacity, etc. Individual tags denote that each spool has unique properties and each has a designated destination in the facility under construction as shown in the project documents. Spools are subject to inspection before leaving the fabricator's plant. The outcome of the inspection activity is that a spool will be found fit-for-installation with an $x\%$ likelihood, and, thus, that there will be a problem with $1 - x\%$ of them. In the latter case, the fabricator must rework this spool to rectify the problem, prior to shipping it to the project site.

Concurrently with this off-site materials handling process, construction is under way on site. Roads are built, temporary facilities are brought in, foundation systems are put in place, structural steel is being erected, etc. Crews of various trades must complete their work in each area where spools are to be hung, prior to spool installation. When a specific set of ready-for-installation spools is available on site, and all prerequisite work in the matching area has been completed, then spools can be installed. Completion of this area's installation work then signals to other trades that subsequent work can start.

5. INDUSTRY PRACTICE

Construction practitioners in the process industry know that their work is plagued by uncertainties. In a most interesting study conducted for the Construction Industry Institute, Howell and Ballard (1995) describe prevailing methods for managing such uncertainties in the piping function. "Piping success requires minimizing the extent and effects of uncertainty during fabrication and installation. At present, uncertainty in the timing of deliveries of intermediate products from one continuing activity to another defines the production planning and management problem. Lacking tools to minimize the uncertainty in these flows, managers strive for flexibility so that the project can proceed in the face of erratic deliveries and unexpected problems. On piping extensive projects, they rely on buffers to assure progress despite variations in the timing, sequence, and quality of resources from upstream suppliers. Buffers dampen the effects of variations in the flow of resources and allow flexibility in the choice of work." (Howell and Ballard 1995, p. 6).

Howell and Ballard (1995, their Figure 4.2) created a work map to describe common practice for moving pipe spools from engineering through off-site fabrication to erection. They used it to highlight that when engineering falls behind schedule, fabrication will be delayed, thereby also delaying installation work. In addition, they point out that the order in which drawings are provided to the fabricator and the sequencing in which spools are output by the fabrication process may bear little relationship to site needs, therefore requiring resequencing for site delivery provided

that priority information be available. Time delays and out-of-sequence work make the supply of materials to the job site unpredictable. This leads to inefficiencies because work cannot be adequately planned and executed, and thus results in low productivity.

Figure 1 charts commodity curves based on data from 1 out of 24 projects on which Howell and Ballard collected data for their study. This specific project (Project B in Howell and Ballard 1995 p. 73), required the installation of 2080 pipe spools in a 57-week duration, measured from start of engineering end of installation. It was characterized as "Design well established. Rash of client changes late in project." (p. 73). As can be seen in this figure, schedule slippages (deviation of actual from planned) occurred in completing isometric drawings (ISOs) and in fabricating spools. Nonetheless, installation work progressed nearly as planned, despite these slippages and despite "Access was difficult. Of the four sides of unit only one had good access." (p. 73).

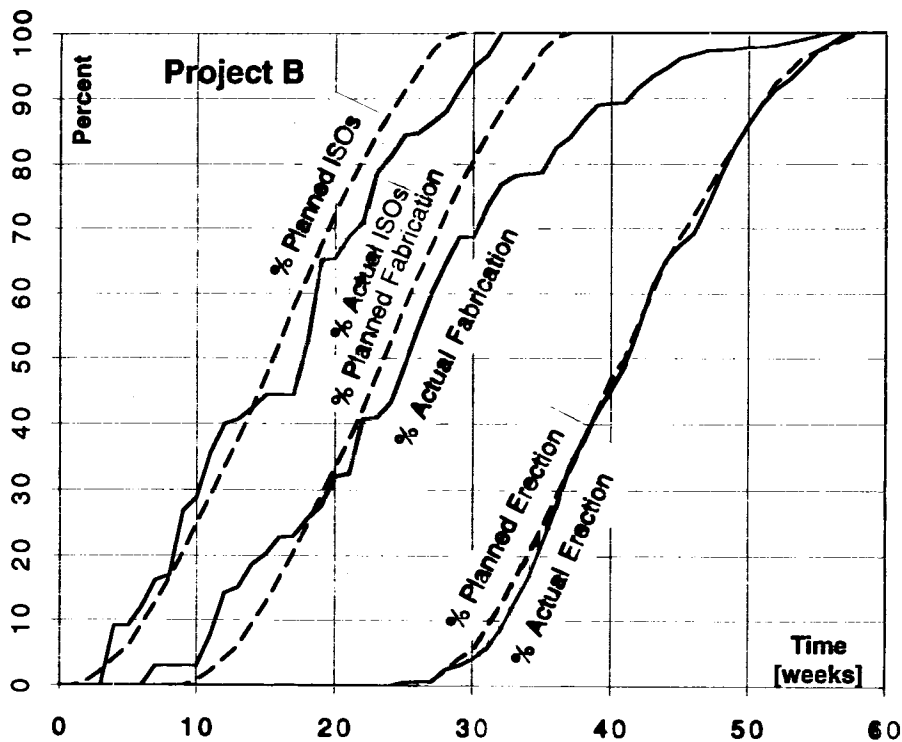


Figure 1: % Planned vs. Actual in Percent of Total of Isometric Drawings (ISOs), Fabricated Spools Delivered to Site, and Spools in Place.

[Data courtesy of Greg Howell and Glenn Ballard]

What is not disclosed by such commodity curves is how much time, money, and effort it took to adhere to the planned installation schedule. Moreover, pipe spools really are not commodities at all. "They differ in specification requirements, physical configuration, and adjacent plant features. These important differences make both establishing performance standards and comparing results very difficult" (Tatum 1985). Howell and Ballard make little additional data available to describe this project's

performance, so one cannot be certain of problem causes and possible solutions. They rate this project's performance at 1.2 (where a value larger than 1 is favorable) but point out "Fast track project. Schedule main concern. Manpower levels above projected." (p. 73). One can only speculate that uncertainty contributed to matching problems, hampering installation work, but apparently successfully overcome.

In conclusion of their study, Howell and Ballard recommend that piping backlogs be used to buffer on-site from off-site activity ("successful projects have at least 60% of all pipe on hand when 20% has been installed"), and that the principles of the "last planner" be applied to shield installation crews from remaining uncertainties.

The present report builds on Howell and Ballard's work by focusing on two uncertainties in the pipe-spool supply chain: (1) uncertainty in duration of the fabrication and transportation process and (2) quality failure in the fabrication process, which results in a delay to rework spools before they can be shipped. It shows how matching affects the productivity of installation crews and the overall project duration. Fictitious data is used, but estimates were taken from Howell and Ballard (1995) where available.

6. PROCESS MODELING








6.1. Process-Model Representation

In order to describe and then experiment with alternative planning sequences for materials management and production control, the pipe-spool installation process has been modeled using the STROBOSCOPE computer system for discrete-event simulation (Martinez 1996). Table 1 summarizes the functionality of the STROBOSCOPE symbols that are used in this example. Note however that their simplicity belies the expressiveness of the STROBOSCOPE programming language associated with them.

One major feature of STROBOSCOPE is that resources can be characterized and individually tracked as they reside in various network nodes during a simulation run. When a queue's resources are indistinguishable, there is only one way in which to draw them from that queue, i.e., only one draw sequence exists. However, when a queue has n distinguishable resources, $n!$ draw sequences are possible. In general, of course, the number n will change in the course of a simulation run as resources join the queue (unless the queue is a source) and leave it (unless the queue is a sink). Being able to distinguish resources and to draw those needed when needed is necessary when one sets out to model matching problems and pull techniques.

The sequence in which characterized resources will be drawn from a queue during simulation depends on (1) the ordering of incoming resources relative to those already in the queue, and (2) the criteria applied in selecting resources for withdrawal from the queue. To achieve the desired system behavior, a STROBOSCOPE programmer can accordingly define draw sequences by specifying (1) a queue's so-called DISCIPLINE and (2) conditions on the link emanating from the queue (e.g., using programming statements such as RELEASEORDER and DRAWWHERE with FILTER-expressions). [Items in CAPS denote STROBOSCOPE programming statements.] Example draw sequences (implemented by ordering, selection criteria, or a combination thereof) are:

Table 1: Selected STROBOSCOPE Symbols

SYMBOL	NAME	EXPLANATION
	Queue	Is a holding place (buffer) for 0, 1, or several resources waiting to become involved in the succeeding combination activity. Queues may contain generic or characterized resources. The latter are distinct from one another and they can be traced as individuals through various network nodes during simulation. The logic describing the ordering of resources upon entry into a queue of characterized resources is termed a DISCIPLINE.
	Normal (activity)	Describes a certain type of work to be done, or a delay, of a known (probabilistic) duration from start to finish. May require a single resource or no resource at all.
	Combi (-nation activity)	Like a normal, describes a certain type of work to be done, or a delay, of a known (probabilistic) duration from start to finish. Unlike a normal, requires several resources in combination for its performance and draws what is needed from the queue(s) that precede it.
	Consolidator	Acts as a counter up to n (n is an integer value specified with the node): after n resources have been released into the consolidator, the consolidated set will be released from it.
	Link	Shows flow logic and should be labeled to meaningfully describe the resources that flow through it. If the link emanates from a queue, a DRAWORDER may be specified to sequence resources being drawn from the queue.
	Fork	Describes a split in a resource's flow path: incoming resources are routed along one path or another in a probabilistic or deterministic fashion. The node is respectively called a probabilistic fork or a decisionnode. Each link emanating from it carries a likelihood or a statement evaluating to true/false for being followed by any specific resource arriving at the fork during simulation. The resource's actual path is determined at run time.
	Assembler	Shows that two or more resources are being assembled into a single unit resource which is of the compound (a special kind of characterized) resource type. The assembly base (if there is one) is marked by a double arrowhead.

1. First-In First-Out (FIFO) or Last-In First-Out (LIFO):

The ordering criterion is the resources' time of arrival in the queue. First-in-first-out or FIFO places resources with earlier arrival time (they have been waiting for the longest time) at the front of the queue, so by default they will be drawn first. In contrast, last-in-first-out or LIFO places those with later arrival time at the front.

2. First-in-Order Based on a Property of Resources in Single Queue:

When resources can be ordered according to one of their properties or based on some other externally-defined numbering system, that order can serve as the

basis for defining draw and subsequent processing priority. For example, trucks of varying size can be sorted by their loading capacity, where it may have been decided that larger ones will be loaded first; a project engineer may have numbered concrete footings at a building site to specify the order in which concrete is to be placed, where lower numbers will be placed first. Selection is thus based on comparing an individual resource's "capacity" or "placement number" property with that of others in the queue, where those with the lower (or higher) number get drawn first. Other examples are "easiest to install first" and "highest ratio of earned to expended effort first" (Howell and Ballard 1995), "materials covered by or buried in others first," "those that can easily be damaged last," etc.

3. Best Match Based on Properties of Resources in Multiple Queues, all Preceding a Single Activity:

Resources may be drawn from several queues so that the properties of those drawn from one queue match those drawn from the other queue(s). In the worst case, no matching resource is available. This situation typifies matching problems. For example, a water boiler will have a designated location in the house under construction, so that when the boiler is ready to be installed, the workers must have access to that designated location to get their work done; no other location will do. Another example is to install only complete pipe runs (Howell and Ballard 1995) or to load various structural steel shapes onto flatbed trucks not to exceed the truck's dimensions or load capacity (Martinez 1996).

4. Random:

Resources are picked at random from those in the queue. This will be appropriate when resources are interchangeable. However, when resources are not interchangeable, random drawing resulting in erroneous substitution (e.g., due to mis-identification of a material) will obviously causes problems.

On any one construction site, one or several of these draw sequences will be used by management and field personnel in the process of handling materials and sequencing work. As is to be expected, the use of some will be more advantageous than that of others. This will be illustrated next by simulating three alternatives. The first two reflect the traditional, push-driven way of sequencing construction work; the third reflects the effect of pulling.

6.2. Pipe-Spool Process Model

A model of the pipe-spool process, described in the example scenario, is depicted in Figure 2. Durations of activities are shown in "days." The rationale for selecting each modeling element and its associated parameters is given below. Admittedly, these selections were somewhat arbitrary despite the author's attempt at being realistic. She modeled salient features of the pipe-spool installation process, yet represented only a small number of system characteristics so that the model's behavior and output would remain tractable. Simplifications were introduced to be consistent with the aim of this

report, which is to illustrate the impact uncertainty has on the execution of resource-matching processes and the benefits of pull-driven over push-driven processing when uncertainty is high, rather than to realistically model any specific project's pipe-spool installation process. The materials flow process varies substantially from one project to the next and virtually no data has been published on any one project with detail on how it was carried out. Also, there appears to be no standard industry practice in this regard and a project-specific model may not be of relevance to the readership at large.

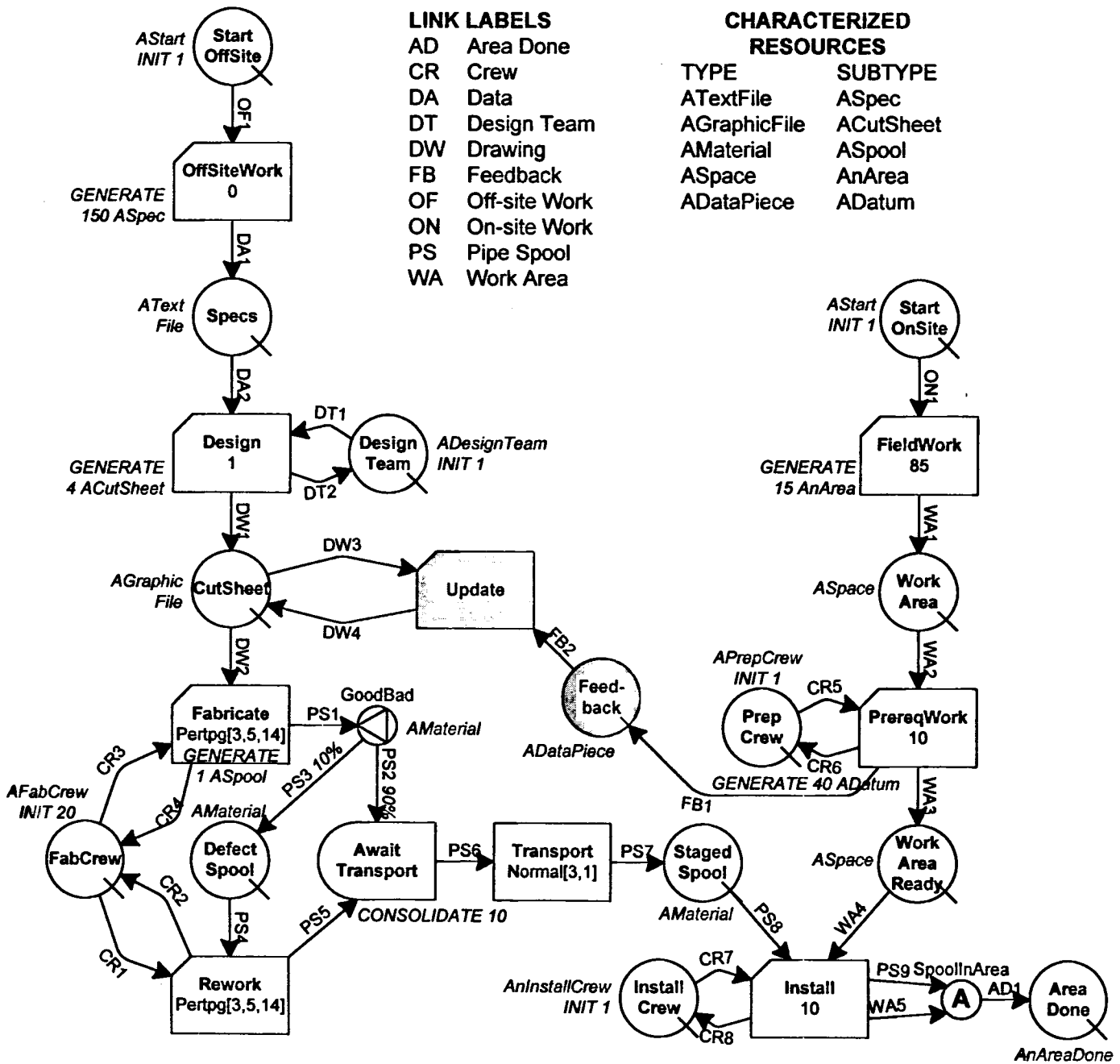


Figure 2: STROBOSCOPE Process Model of Pipe-Spool Installation

6.2.1. Off-Site Activity Sequence

StartOffSite [Queue]: This node defines the start of all off-site work. The point in time at which a resource in this queue becomes available is decided by the project owner. In the present simulation model, this queue is initialized to have one resource at time 0 (zero).

OffSiteWork [Combi]: Off-site work comprises many activities, such as defining the project scope; obtaining permits; selecting architect/engineers (AEs), vendors, and contractors; awarding contracts; etc. These are all combined into one OffSiteWork activity with an arbitrarily-chosen, deterministic duration to keep the model simple. In this way, uncertainties regarding these early project steps will not obfuscate the remainder of the model though, in reality, they often are major.

A gross simplification was introduced by assuming that upon completion of OffSiteWork, 150 Specs are generated. These in turn will be processed one at time by the DesignTeam. No industry data was available to support any more realistic assumption.

The start and duration of the OffSiteWork activity in this first activity chain must be coordinated with that of the FieldWork activity in the second activity chain. Typically, process plants are constructed according to a fast-track schedule, so field work will start before completion of design. Start and duration parameters were thus chosen accordingly: at simulation time, StartOffSite will release one resource at time 0 for OffSiteWork to immediately start, and the OffSiteWork duration is set at 0 days.

Specs [Queue]: OffSiteWork results in project scope definition and specifications (Specs) to be used by the design team.

DesignTeam [Queue]: The design team represents the group of architects and engineers that design and engineer the facility.

Design [Combi]: Design includes performing preliminary design studies, detailing process-, structural-, and piping systems, as well as preparing contract documents.

In the process industry, designers document the system configuration by means of isometric drawings (ISOs) or piping and instrumentation drawings (P&IDs) showing pipe routing, valves, instrumentation, etc.

Industry practice varies on whether it is the designer or the fabricator who uses these drawings to engage in "spooling": long pipe runs shown on a single ISO must be cut into pieces to allow for off-site prefabrication and easy shipment to the site, followed by field-assembly during installation. Spooling results in fabrication drawings called "cutsheets"—typically there are on the order of 4 cutsheets per ISO (cf. Michael Villejas, personal communication)—each one detailing a single spool and assigned a number with which the spool will be tagged and tracked throughout the piping process.

The 1-day duration for Design was chosen somewhat arbitrarily, but when combined with the single DesignTeam, it yields a production rate of 4 cutsheets/day. This rate matches the most likely duration production rate of Fabricate, PrereqWork, and Install.

CutSheet [Queue]: Once cutsheets are available, spool fabrication can start.

A new process facility may comprise several hundreds or thousands of spools,

which are to be installed over the duration of several months or years of construction. The present model includes 600 spools (i.e., 150 specifications generated by OffSiteWork, each resulting in 4 cutsheets upon completion of Design).

FabCrew [Queue]: As shown in Figure 2, the fabricator's only resource constraint being represented symbolically is the availability of fabrication crews. Of course, a production schedule will reflect the fabricator's ability and intent to fabricate spools at a supply rate agreed-upon with the owner or contractor of the project at hand. This schedule presumably takes into account the fabricator's plant capacity, stock material on hand, work load, machine maintenance schedules, etc. To an outsider, however, details of this schedule usually are not known. Only the delivery schedule meeting the supply rate agreed upon by the owner, contractor, and supplier, will be communicated to those responsible for procurement and (possibly) on-site receiving.

20 fabrication crews are available so that the expected throughput of Fabricate (with a most likely duration of 5 days per spool) will be 4 spools/day.

Fabricate [Combi]: The duration of fabrication of a pipe spool is characterized by a skewed distribution with a most likely value of 5, an optimistic value of 3 and a pessimistic value of 14 days as was suggested by Howell and Ballard (1995 p. 19). These values are best approximated using a PERT distribution (Perry and Grieg 1975). Using a single distribution to describe fabrication of all spools factors out several process parameters and uncertainties: for instance, one would expect the duration to vary with the quality of the drawings and specifications, the complexity of the spool's design relative to the machining skills of the fabricator and available tools, and the uncertainty of the fabrication process (machine accuracy, precision, down time, etc.) and the production schedule.

GoodBad [Fork]: This probabilistic fork helps model variation in fabrication quality. In the process industry today it is common for the rework rate on pipe spools to range from 1 to 10% (Ballard 1996). The PS3 link emanating from this fork and pointing to the DefectSpool queue, which describes the path of a defective spool, will therefore have an $x\%$ likelihood of being followed by any one spool, where $1 \leq x \leq 10$. The alternative path, shown by the PS2 link pointing to the AwaitTransport consolidator, has the complementary likelihood of $1 - x\%$.

Numerous other quality failures could exist in the piping process. They could have been modeled by means of similar forks following normal or combination activities in Figure 2, but this was not done for the sake of simplicity.

DefectSpool [Queue]: Defective spools denote work in progress. They wait for rework to be done by the fabricator's resources (FabCrew) as soon as these become available. When there are defective spools, FabCrew will have to decide on which activity to do first: Rework or Fabricate. It has been assumed here that Rework takes priority as this will lead to minimal delay in transporting spools that may have gotten out of sequence.

Rework [Combi]: Rework is done to rectify quality failure during spool fabrication. Due to lack of other data, its duration distribution is assumed to be the same as that of Fabricate. Though this is a bad situation, it is by no means the worst possible. For instance, when a quality failure has been attributed to an inconsistency in the design drawings, it may take a significantly longer time to get this issue clarified through a request-for-information cycle.

AwaitTransport [Consolidator]: In practice, a load can comprise anywhere from one to tens of pipe spools, the actual number being a function of for instance demand, spool geometry, truck dimensions, loading constraints, and site delivery requirements. Here, it is assumed that each flatbed truck is loaded with ten spools in the order in which they leave fabrication, and waits until it is fully loaded before departing. In addition, a truck is always available when one is needed, so few spools will accumulate at the fabricator's site.

More realistic had been to show a buffer of spools accumulating to allow the fabricator to re-sequence spools and optimize loads before shipment. Re-sequencing is necessary when a ordered set of spools has been specified for site delivery, and, due to uncertainty prior to or in the fabrication process (possible necessity for doing rework and variation in duration of Fabricate and Rework, but generally-speaking also variation in duration of OffSiteWork and Design), spools do not exit fabrication in that order. "Differences between the input and output sequences may be due to simpler spools passing more complex ones on the line, spools losing place in line because of quality failure recycle, and the difference in demand for, and rate of production by lines dictated to certain size ranges" (Howell and Ballard 1995 pp. 19-20). In the present model, this buffering and re-sequencing at the fabricator's end has been consolidated with buffering and re-sequencing in the StagedSpool queue on site, necessary prior to spool installation.

Transport [Normal]: The duration of transportation is assumed to be normally distributed, requiring three days on average plus or minus one day. Again, this distribution factors out several process parameters and uncertainties, such as traffic delays encountered en route or stops at multiple sites prior to offloading at the destination site.

StagedSpool [Queue]: Finally, spools are off-loaded at the project site. On-site handling usually is a staged process, including receiving, long-term laydown, issuing, rehandling, and short-term staging before final installation. For simplicity's sake, only a single staging buffer has been modeled.

6.2.2 On-Site Activity Sequence

StartOnSite [Queue]: This node defines the start of all on-site work. Exactly when ground breaking takes place is determined by the project owner and other parties in the AEC process. Again, as was the case for StartOffSite, in the present simulation model, StartOnSite is available at time 0 (zero).

FieldWork [Combi]: Field work comprises many activities, such as clearing the site, excavating, installing foundation systems, and erecting structural steel before any pipe spools can be hung. These are all lumped into one activity with a deterministic duration to keep the model simple. In this way, uncertainties regarding these early project steps will not obfuscate the remainder of the model though, in reality, they often are major.

Since the model is to represent a fast-track project, it has been assumed that FieldWork takes 85 working days (roughly 17 weeks or 4 months) to complete after OffSiteWork is done and project specifications (Specs) have been generated. This

duration is on the lower end but within the range of values listed as the duration between Engr-to-Fab-Plan and Fab-To-Install-Plan on projects detailed in Howell and Ballard (1995 pp. 69-95). Upon completion of FieldWork, 15 Areas are generated.

WorkArea [Queue]: Output from FieldWork is space in the facility under construction where subsequent work is to take place. This space is divided into work areas which help define work packages for project planning.

PrepCrew [Queue]: As was the case for FabCrew, only a single resource constraint is being represented here, though it symbolizes a combination of resources, such as a construction schedule, equipment, materials, and space availability.

PrereqWork [Combi]: The work that immediately precedes the installation of pipe spools is so-called prerequisite work. In this model it is assumed that each area's prerequisite work must be completed in its entirety before an installation crew can move in and install the corresponding spools. In addition, it is assumed that 40 specific spools are to be installed per area, so with a PrereqWork duration of 10 days per area, this matches the expected production rate of 4 spools/day (also see Design, Fabricate, and Install).

WorkAreaReady [Queue]: Output from PrereqWork is space in the facility under construction where pipe spools will be hung next.

The gray-shaded symbols Feedback [Queue] and Update [Combi] are explained later.

6.2.3. Joint Activity Sequence

InstallCrew [Queue]: Again, only a single site resource constraint is being represented here (namely an installation crew in InstallCrew) though it symbolizes that a combination of resources is needed, not only pipefitters, but also installation drawings, a weekly work plan, and tools and equipment. As for the Install work that this crew performs, constraints on materials (pipe spools in StagedSpool) and space availability (work areas in WorkAreaReady) have been modeled explicitly.

Install [Combi]: Each area opened up to pipefitters requires a number of spools to be installed as is defined in the corresponding work package. As mentioned previously, in the present model areas are assumed to require 40 specific spools each so with an Install duration of 10 days per area, this matches the expected production rate of 4 spools/day (also see Design, Fabricate, and PrereqWork). Install will not start until all 40 spools are available in StagedSpool and the matching area is in WorkAreaReady.

SpoolInArea [Assembler]: When the appropriate 40 spools have been installed satisfactorily in a work area, that area has been completed with respect to the work being modeled here. This completion is symbolically represented by a compound resource made up of an assembly of two resources, the specific area and the 40 matching spools.

AreaDone [Queue]: The compound unit resource accumulates in the AreaDone queue, whose elements can be monitored to assess percent completion of the installation process or resource availability for subsequent work that is not modeled here.

7. IMPLEMENTATION AND SIMULATION OF ALTERNATIVE MODELS

Using Figure 2's graphical representation of the pipe-spool process, one deterministic and three probabilistic models were implemented. Their process execution has been simulated using STROBOSCOPE. Simulation outputs have been charted for a single run to show what a specific instance of each model might look like, and for 1000 iterations to obtain average data values that better characterize each model.

Source code input to STROBOSCOPE for the models is provided in Appendix 2. Note that the use of different draw orders can be discerned in the source code but not in Figure 2 as the graphical representation reflects only a limited number of model parameters.

7.1. Deterministic Model

7.1.1. Model Parameters

The deterministic model is based on the process chart with all characteristics shown in Figure 2 except for those listed in Table 2. This model illustrates how the project progresses in case there would be no uncertainty in the system. If activity durations took on their most likely value, if there were no quality failures (no rework), and if all off-site and on-site resources were perfectly sequenced, then—by selection of the model's variables—the system would be synchronized. Perfect sequencing means that cutsheets 1 through 600 will be used for fabrication in numerical order, resulting in spools 1 through 600 arriving at the site in numerical order. Similarly, work areas 1 through 15 will be prepared in numerical order, so that installation of spools 1 through 40 starts without delay as soon as area 1 is ready, installation of spools 41 through 80 starts as soon as area 2 is ready, and so on. By construction, all CutSheets are available on day 0, at the start of simulation, and WorkAreas on day 85. The durations of activities Design, Fabricate, PrereqWork, and Install, combined with their production resources DesignTeam, FabCrew, PrepCrew, and InstallCrew respectively, were chosen so that their throughput is equivalent to 4 spools/day. Accordingly, synchronization means that the commodity curves (show on a chart that plots percent complete vs. time, also called a line-of-balance or velocity chart) for CutSheet, StagedSpool, WorkAreaReady, and AreaDone are parallel to one another (Figure 3).

Table 2: Deterministic vs. Probabilistic Models

MODELING SYMBOL	DETERMINISTIC MODEL (Model D)	PROBABILISTIC MODEL (Models A, B, and C)
STRENGTH PS3	0% (no rework)	10%
STRENGTH PS4	100%	90%
DURATION Fabricate	5	Pertpg [3, 5, 14]
DURATION Rework	N/A	Pertpg [3, 5, 14]
DURATION Transport	3	Normal [3, 1]

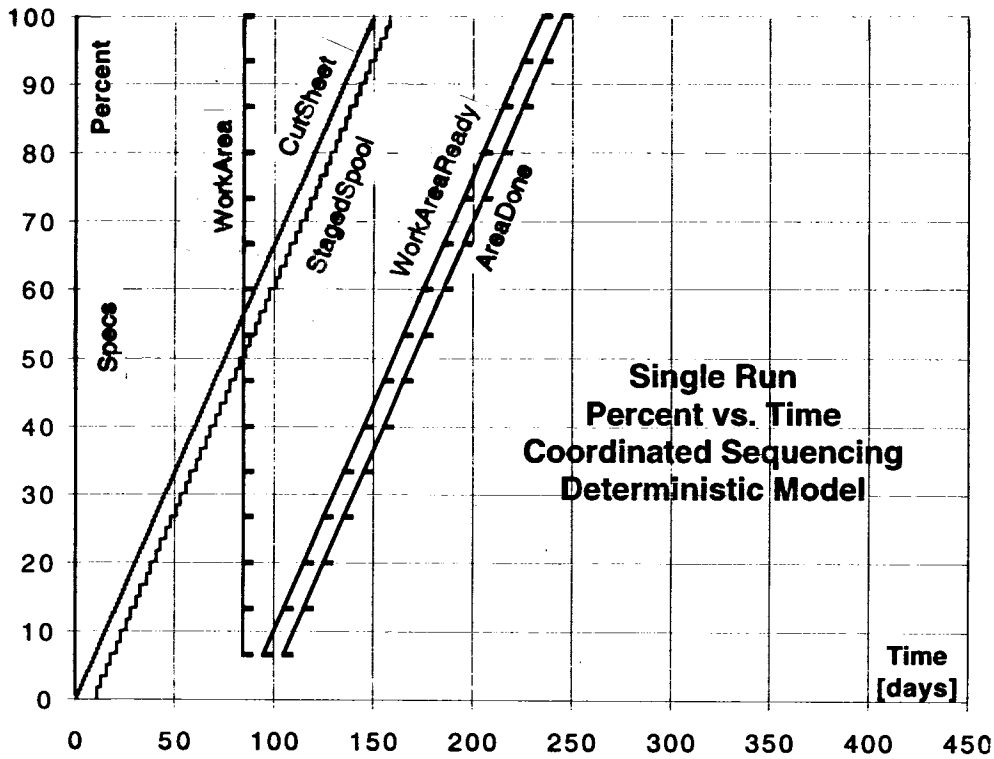


Figure 3: Single Run Percent Complete vs. Time for Coordinated Sequencing in Deterministic Model

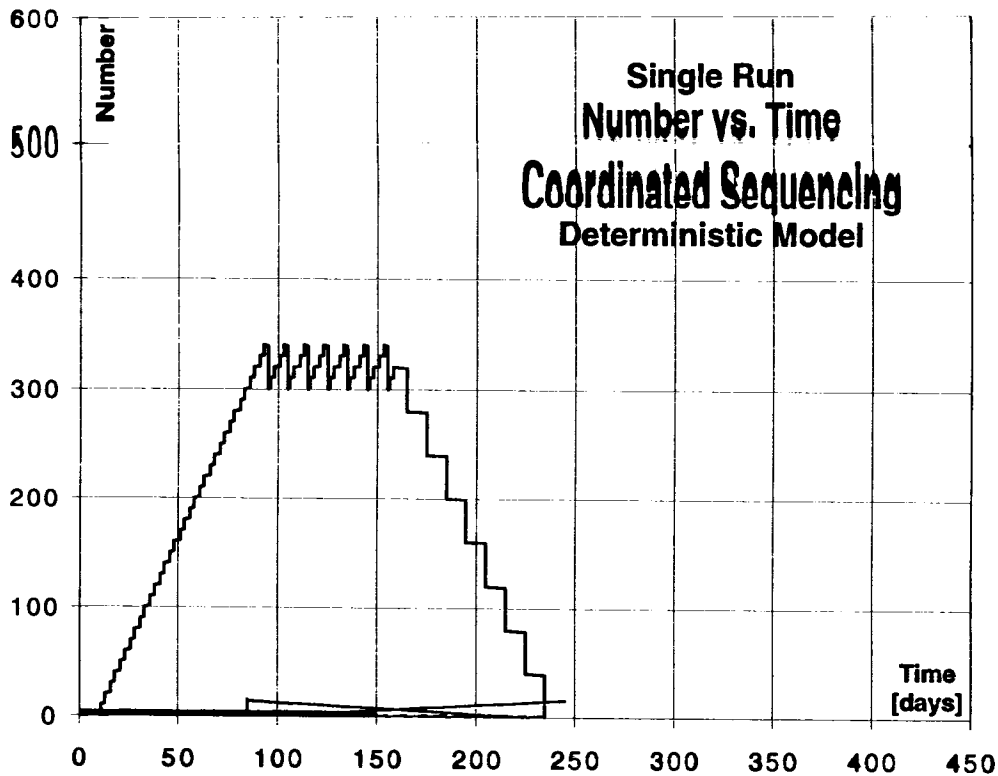


Figure 4: Single Run Number Resources vs. Time for Coordinated Sequencing in Deterministic Model

7.1.2. Pipespool Buffer Size

Figure 4 illustrates that pipe spools accumulate on site at a steady pace until day 95, when the first area opens up for installation. The StagedSpool buffer peaks at 340. It gradually gets depleted over time as subsequent areas open up but continues to get replenished until all spools have been delivered to the site.

Note that at 20% of work done (AreaDone), nearly 80% of the total number of spools have been delivered to the site. According to Howell and Ballard's rule-of-thumb, this is a necessarily condition for project success.

7.1.3. Productivity of Installation Crew

Since all activities in this model are synchronized and no uncertainties manifest themselves, the installation crew's productivity is optimal. Each area gets completed in 10 days and the crew can proceed without being interrupted due to lack of materials or space to work.

7.1.4. Project Duration

The project duration is 245 days (95 days until work in the first area can start plus 15 times the installation of 40 spools in each area where work progresses at 10 days/area: $95 + 15 \cdot 10 = 245$).

7.2. Alternative Probabilistic Models

7.2.1. Model Construction

The three probabilistic models also are based on the process chart depicted in Figure 2, but they do include all uncertainties in terms of activity duration and likelihood of rework as shown. They differ from one another in two ways: (1) they use a different order in which to draw cutsheets for fabrication of pipe spools and (2) only the last model includes the Feedback queue, the Update activity, and links FB1, FB2, DW3, and DW4 (see Figure 2).

Other draw sequences or feedback mechanisms could have been implemented and their impact studied on, for instance, crew productivity or project completion. Readers interested in reproducing the outputs presented here or running other scenarios can refer to Appendix 2 which lists the computer source code.

In the probabilistic models the duration variables were chosen so that the *most likely* or *mean* value of the off-site activity sequence is synchronized with the on-site sequence. This means that the most likely or mean values of DesignWork, Fabricate (ignoring Rework), and Transport add up to have the right number of pipe spools—though not necessarily the right spools—being staged on site for a desired number of days before the mean value of completing FieldWork and PrereqWork. In addition, the throughput (average number of resources output per time unit) off-site matches the throughput on-site (40 spools get produced on average in the same amount of time needed to complete prerequisite work in an area). That way, presumably (if all instances of each activities had a duration close to the mean value of that activity and no quality failure such as rework manifested itself), field production should not be delayed by a shortage of materials. Nevertheless, it could be delayed due to mismatches. Table 3 lists the alternative draw sequences used in each of the probabilistic models (A, B, and C). It also includes that of the deterministic model described previously (model D).

The first two probabilistic models illustrate the traditional, push-driven way of managing a construction process, but they present extremes in effort put into pre-construction planning. The third implementation provides one example of pull-driven process management.

7.2.2. Data Collection and Representation

Draw sequencing does not affect throughput of the fabrication process (by construction of the model), so in terms of percent of pipe spools having been fabricated, transported, and delivered to the site, all three probabilistic models exhibit the same behavior. Figure

5 shows the mean value as well as the mean plus-or-minus 1-or-2 standard deviations for the percent StagedSpool delivered to the site over time for 1,000 iterations.

Data for 1,000 iterations was collected for each of the probabilistic models. This was done using STROBOSCOPE 'COLLECTORS' keeping track of the incoming number and the current count of resources in queues.

Table 3: Alternative Sequencing Strategies

CASE	DESCRIPTION	CutSheet DRAW SEQUENCE	WorkArea DRAW SEQUENCE
A	Probabilistic Model Random Sequencing	Random	FIFO
B	Probabilistic Model Coordinated Sequencing	FIFO	FIFO
C	Probabilistic Model Pull-driven Sequencing	Priority to spools that match area(s) ready	FIFO
D	Deterministic Model Coordinated Sequencing	FIFO	FIFO

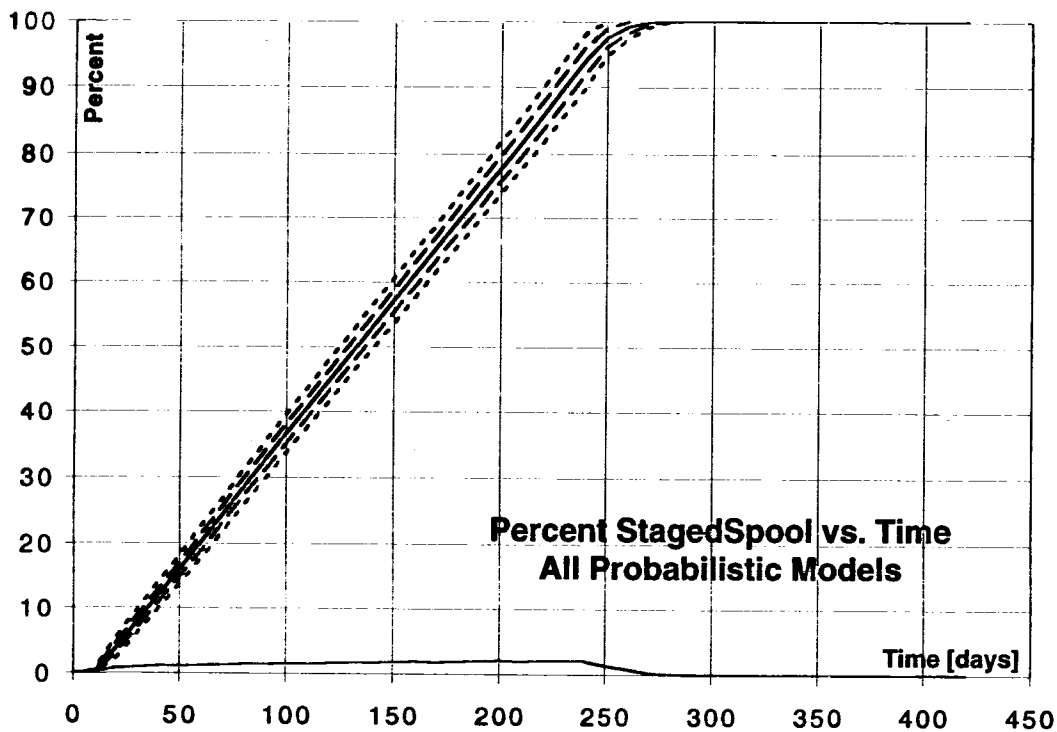


Figure 5: 1000 Iterations Percent StagedSpool vs. Time for All Probabilistic Models

7.2.3. Model A - Random Sequencing

7.2.3.1. Model Parameters

This first model describes a worst-case scenario as far as pre-construction planning is concerned: there is no coordination at all! The order in which pipe spools are fabricated bears no relationship to the order in which areas are being prepared on site. With no communication between fabrication and installation to coordinate their respective work with one another, each crew will draw resources from the queue available to them in the order that suits them best. The fabrication crew may select spools based, perhaps, on the order in which cutsheets are provided to the fabrication shop. This order may have nothing to do with installation sequencing. The preparation crew may select one area after the other based, perhaps, on which one is adjacent to their present location. This situation has been implemented by defining the DRAWORDER for choosing resources in the CutSheet queue through link DW2 to be random, and leaving the discipline in the WorkArea queue as the default FIFO (anything else would appear equally random relative to the randomness in drawing cutsheets).

7.2.3.2. Pipespool Buffer Size

Because of lack of coordination, the likelihood for mismatches to occur at installation time is high. The problem manifests itself when distinct spools in StagedSpool and work areas in WorkAreaReady need to be matched up for installation. When no match can be found, staging areas will fill up with spools of no immediate use, and work areas remain unfinished and therefore unavailable for subsequent work to take place. The peak at 570 for StagedSpool in Figure 7 confirms this to be the case. (Note that 590 spools on site is the largest number possible in this model, as there are 600 spools in total, 15 areas requiring 40 spools each, and the next-to-last shipment of 10 should at the very latest complete five required sets of 40).

In reality (not in the model), spools on site for a long time (at worst, a spool could be the first one delivered and last one installed) are more likely to sustain damage. Additional preparation work (e.g., erecting scaffolds or, at least, checking if previous preparation work still meets specifications and if there are no new obstructions) may be required when installation crews finally move into an area where work is to be completed. Real project costs are incurred for keeping track of and rehandling materials on site, for working in obstructed areas, and for performing out-of-sequence work.

7.2.3.3. Productivity of Installation Crew

As mentioned previously, it has been assumed (in all four models) that installation in an area does not start until all 40 spools for that area are available. This is not necessarily industry practice, but the line of balance labeled AreaDone in Figure 6 confirms that accumulating a huge buffer makes it possible for the installation crew to achieve its highest possible production rate (also see Howell et al. 1993). At 20% of spools installed, virtually all spools have been delivered to site; vice versa, at 65% of spools delivered,

barely any installation work has been completed. For the crew, this will be an effective way of getting work done, provided that they need not be idle prior to starting work.

In this single run, the crew's exact start time is 246.66 days. However, during planning, one can only estimate their start date. Given the uncertainty in getting matching spools to site (Figure 5), it may make sense to build in a time buffer or lag preceding the Install activity to enable the crew to be optimally productive once they mobilize (e.g., start Install no earlier than day 250). Such shielding (Ballard and Howell 1997) protects the installation crew from upstream process uncertainties.

7.2.3.4. Project Duration

The delay in starting installation work, forced here by a shortage of matching materials, leads to a project duration of 397 days, which is by far the longest one of all models.

Because AreaDone is a sink (no resource ever leaves the queue) a graph showing Percent AreaDone vs. Time and one showing Number AreaDone vs. Time will have exactly the same shape, so only the first one has been included for this and the other probabilistic models. The project duration for 1,000 iterations has been fitted using a normal distribution.

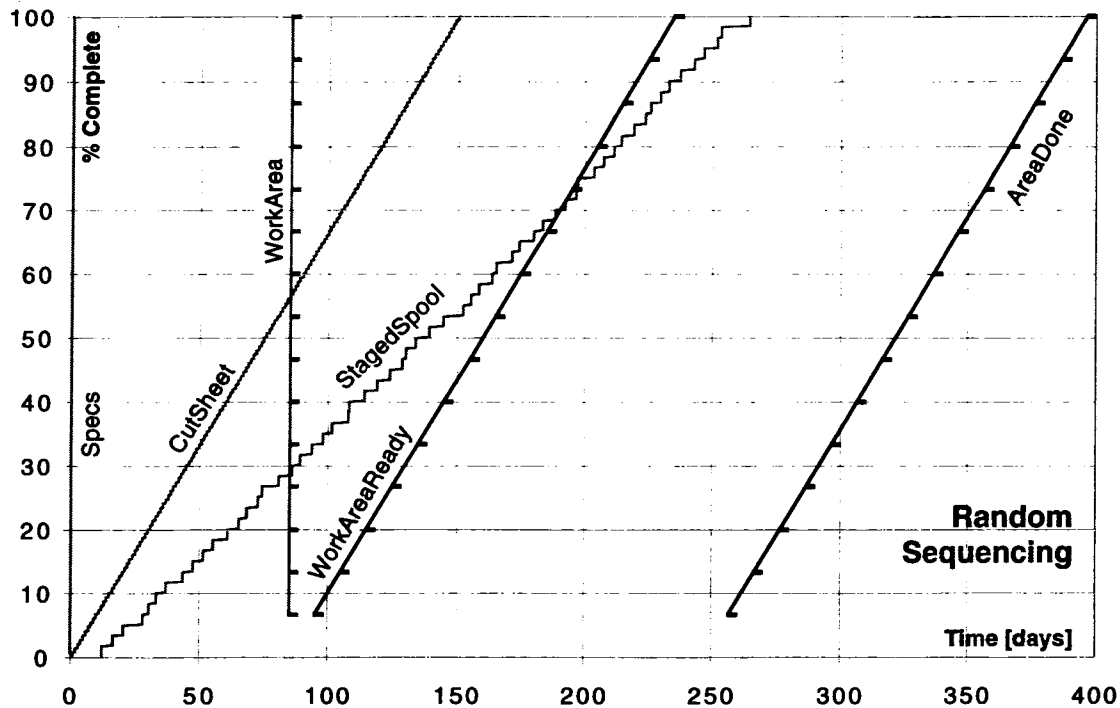


Figure 6: Single Run Percent Complete vs. Time for Random Sequencing

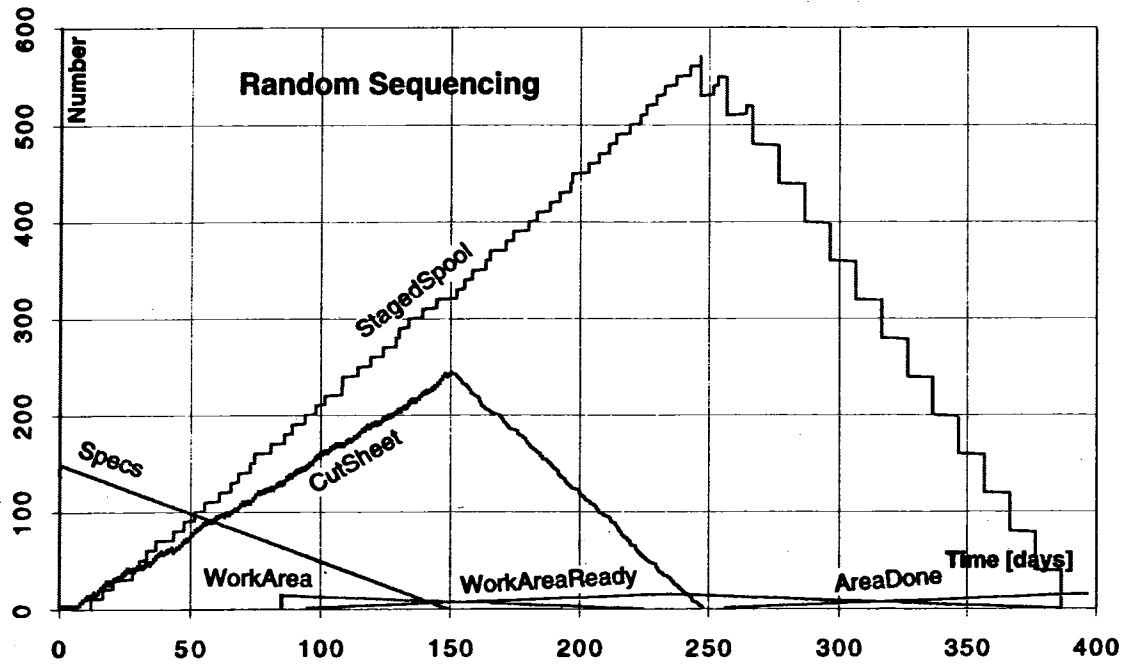


Figure 7: Single Run Number Resources vs. Time for Random Sequencing

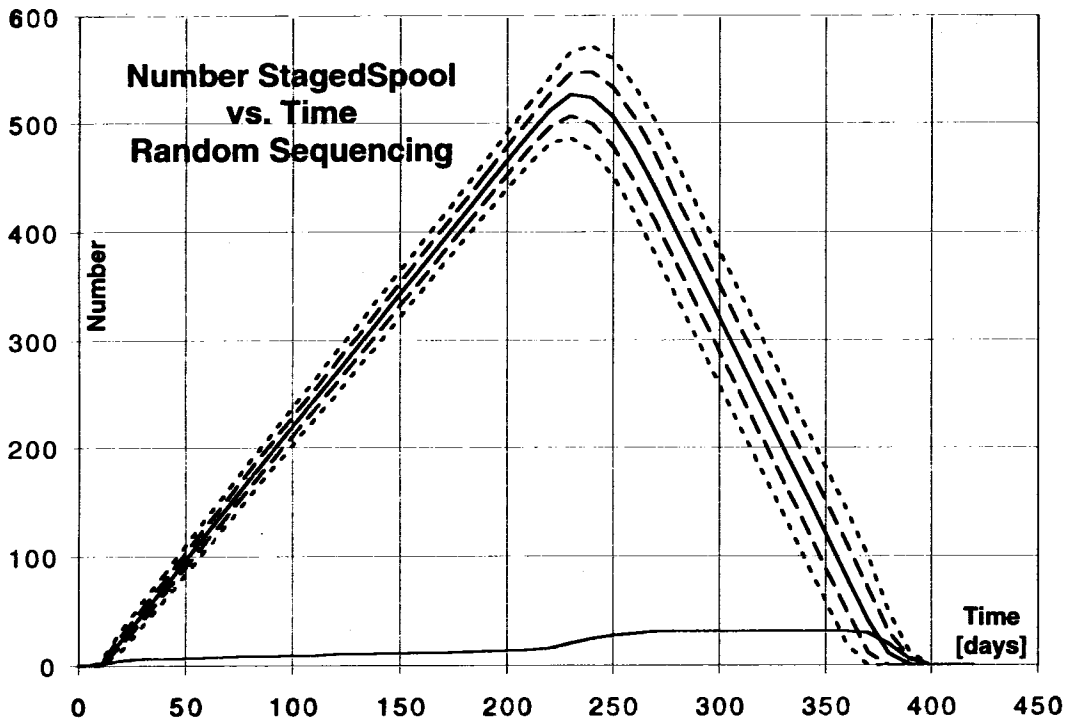


Figure 8: 1000 Iterations Number StagedSpool vs. Time for Random Sequencing

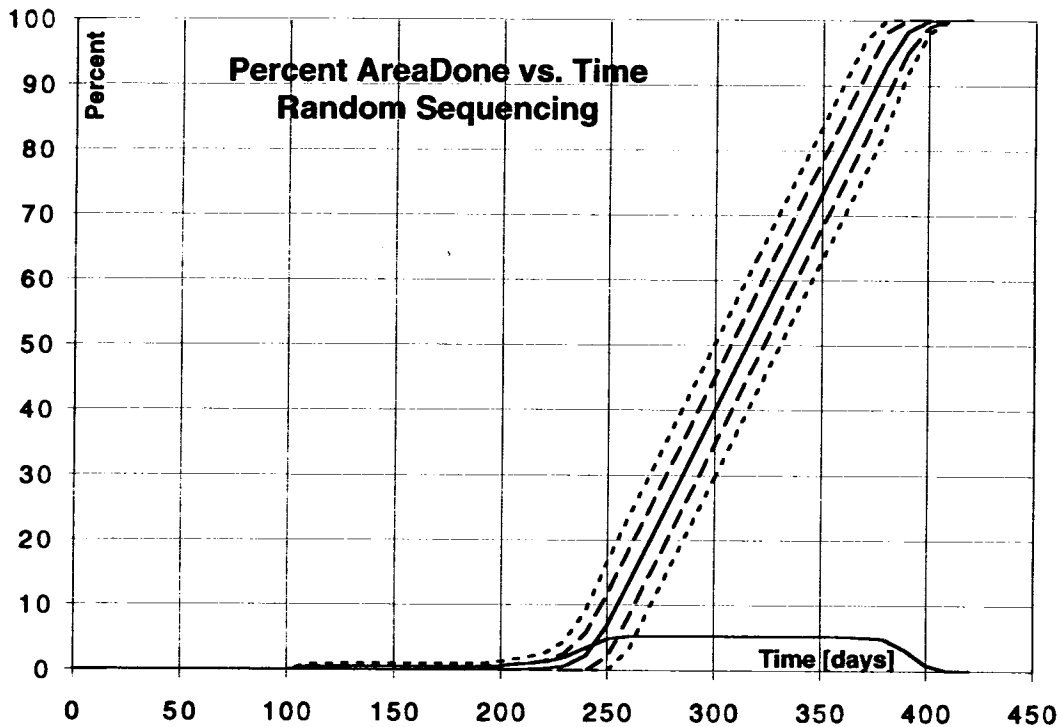


Figure 9: 1000 Iterations Percent AreaDone vs. Time for Random Sequencing

7.2.4. Model B - Coordinated Sequencing

7.2.4.1. Model Parameters

Model B describes perfect coordination. This means that the fabrication crew and the installation crew plan before starting their work and decide on the sequence in which to draw resources. Cutsheets and areas are assigned sequence numbers so they can be drawn in FIFO order (STROBOSCOPE's default discipline). CutSheets 1 through 40 will go to fabrication before 41 through 80, and so on. Similarly, Area 1's prerequisite work will be performed prior to Area 2's, and so on.

While perfect coordination reflects an idealized situation, for many reasons it will never materialize. It is too restrictive to the various parties involved in the process (e.g., fabrication shops are not set up to tolerate one-piece flows, that is, to change machine setups in order to meet each spool's unique fabrication requirements).

7.2.4.2. Pipespool Buffer Size

Model B results in minimal space needed to stage spools on site: StagedSpool peaks at 200 in Figure 11. Nonetheless, some spools will accumulate on site because they get out of sequence due to the manifestation of the uncertainty in the duration of fabrication and shipping, and the occasional need for rework.

7.2.4.3. Productivity of Installation Crew

Despite expedient project completion, the installation crew (which starts to work as soon as work is available and stays idle in-between activities, when materials are in short supply) was not able to work as productively as before (the AreaDone line of balance is not straight but bends to the right). This is no coincidence! The author crafted the model's basic template to show how materials shortages might arise so that their impact on production could be shown. While the activities Design, Fabricate, PrereqWork, and Install can process resources at the same average rate of 1 area/10 days or 4 spools/day, uncertainty in the Fabricate, Rework, and Transport processes results in a StagedSpool slope much smaller than the CutSheet or WorkAreaReady slope. Consequently, the AreaDone slope is smaller as well (note that in Model A the AreaDone slope was virtually not affected by the slowed delivery rate because of the large build-up of spools prior to its start). Because FieldWork starts 85 days (by construction) after OffSiteWork, the StagedSpool and WorkAreaReady lines of balance cross.

7.2.4.4. Project Duration

Perfect coordination leads to project completion in the shortest duration (275 days, Figure 10).

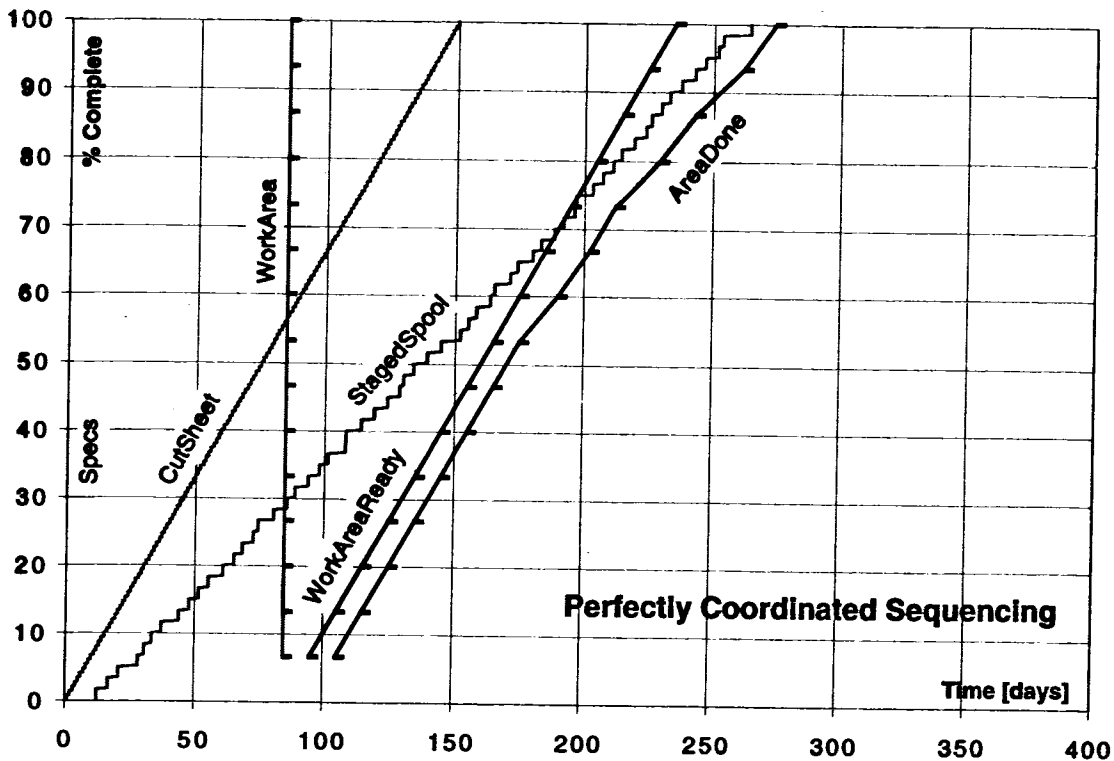


Figure 10: Single Run Percent Complete vs. Time for Coordinated Sequencing

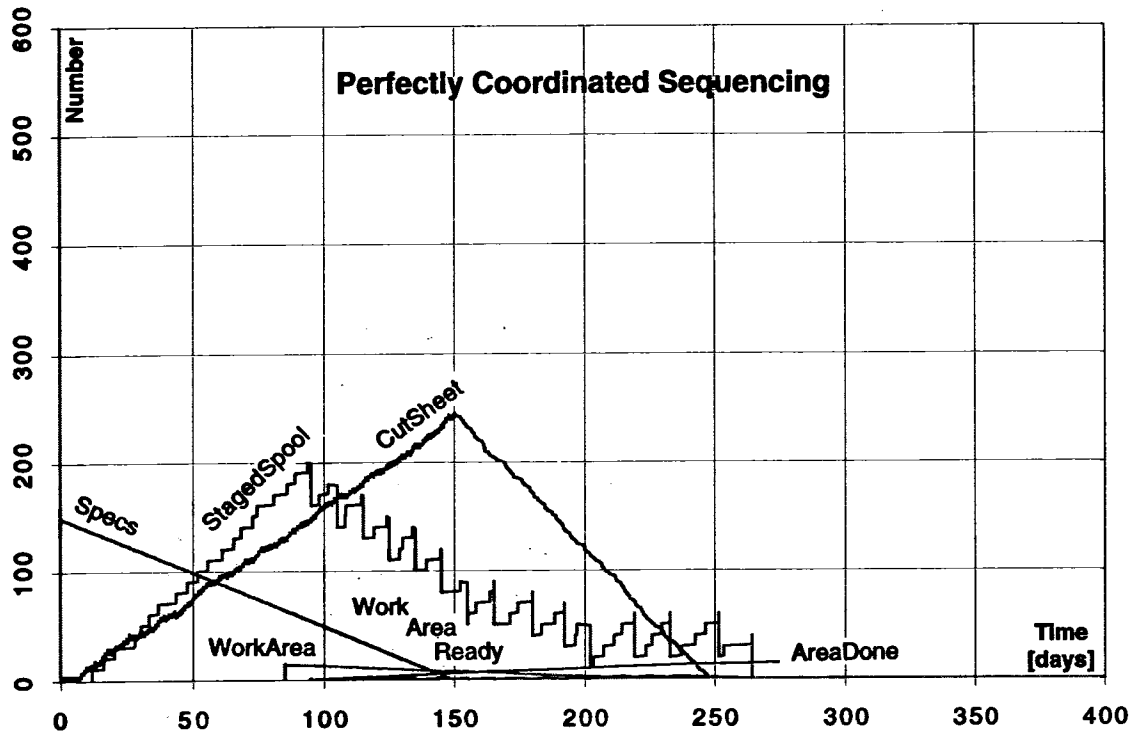


Figure 11: Single Run Number Resources vs. Time for Coordinated Sequencing

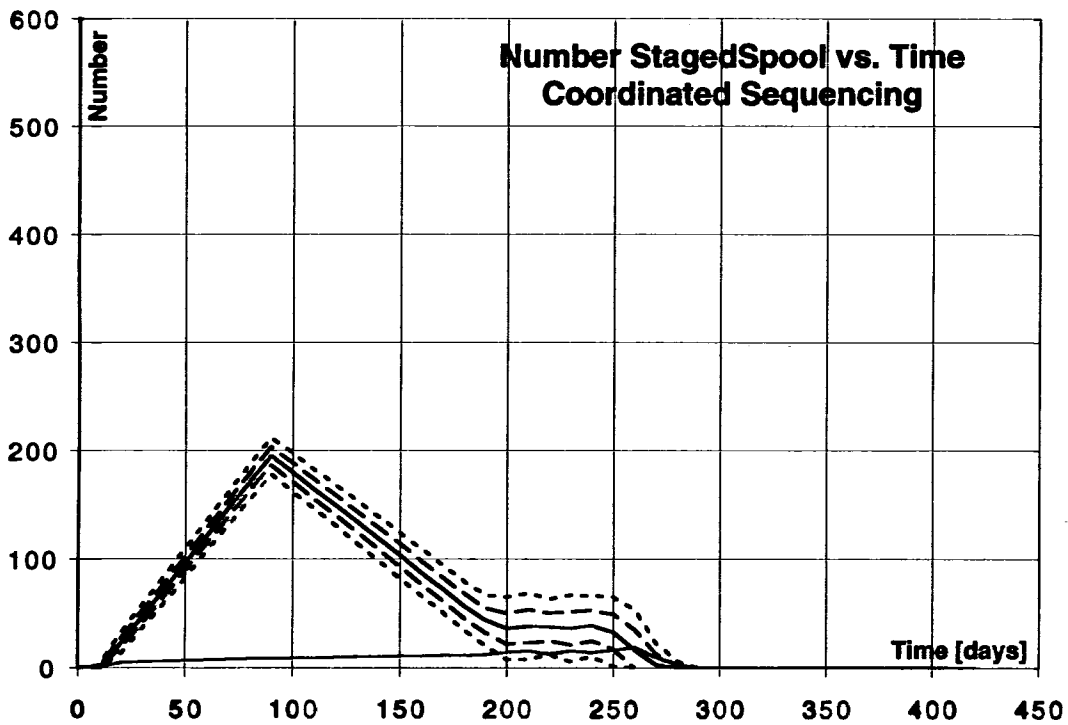


Figure 12: 1000 Iterations Number StagedSpool vs. Time for Coordinated Sequencing

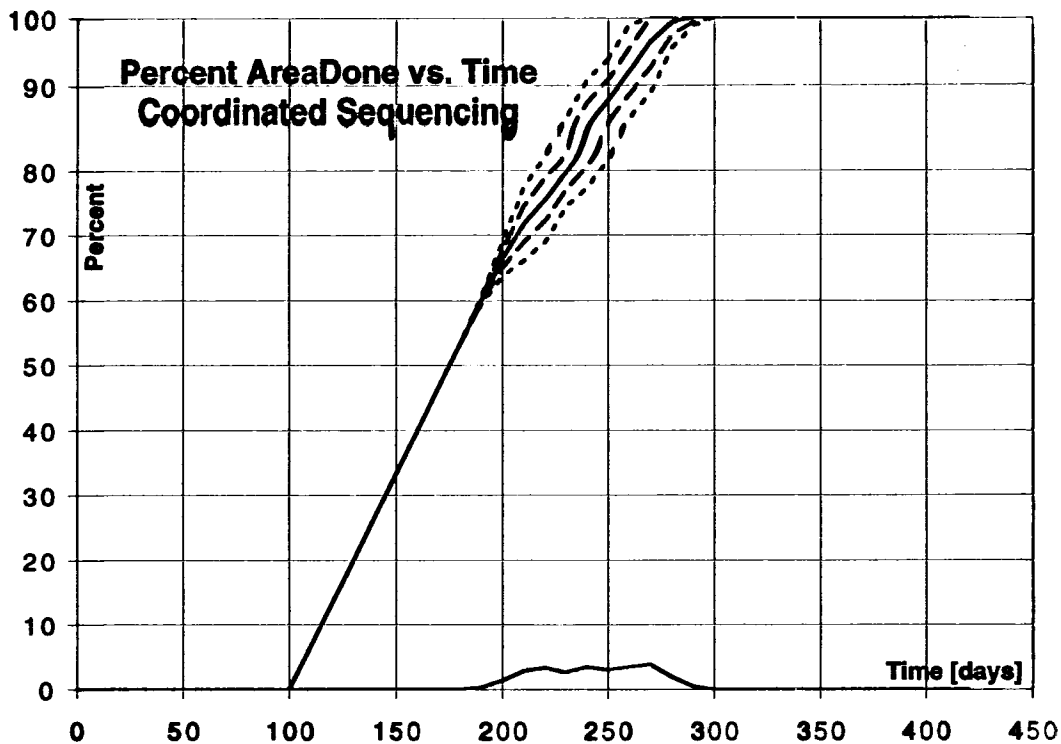


Figure 13: 1000 Iterations Percent AreaDone vs. Time for Coordinated Sequencing

7.2.5. Model C - Pull-driven Sequencing

7.2.5.1. Model Parameters

Model C augments model A's random sequencing with a pull mechanism, which includes the Feedback queue, the Update combination activity, and four links to tie them into the existing network.

Feedback [Queue]: When the PrereqWork combi finishes, 40 pieces of feedback data are released along the FB1 link into this queue. This feedback data describes which area completed and which matching cutsheets should therefore be expedited through the fabrication process.

Update [Combi]: Feedback data is then used to update the status of cutsheets in the CutSheet queue. Those cutsheets that match area-availability feedback are checked accordingly so that they will get priority over others to be fabricated.

This model has been crafted to process CutSheets in random order relative to work areas, but as soon as an area is ready for spool installation, the CutSheets are re-prioritized based on matching feedback. Consequently, they are processed ahead of others, that is, they are "pulled" to the site. In the single-run model, a total of 291 updates were performed.

7.2.5.2. Pipespool Buffer Size

Relatively few spools accumulate on site (250 maximum, Figure 15). The buffer is not as small as it was in the case of perfect coordination, but it certainly does not get out of hand as it did in the case of random sequencing.

7.2.5.3. Productivity of Installation Crew

Starting off with random sequencing and then improving the sequencing based on feedback penalizes the crew in terms of field productivity relative to the perfect coordination case. The slope of AreaDone has decreased further than it already had in model B. Luckily, this can be improved by ordering the crew to start later, when more spools are on site so they will be able to work at their fastest possible rate, or by scaling down crew size.

7.2.5.4. Project Duration

The project duration remains fairly short (304 days, Figure 14).

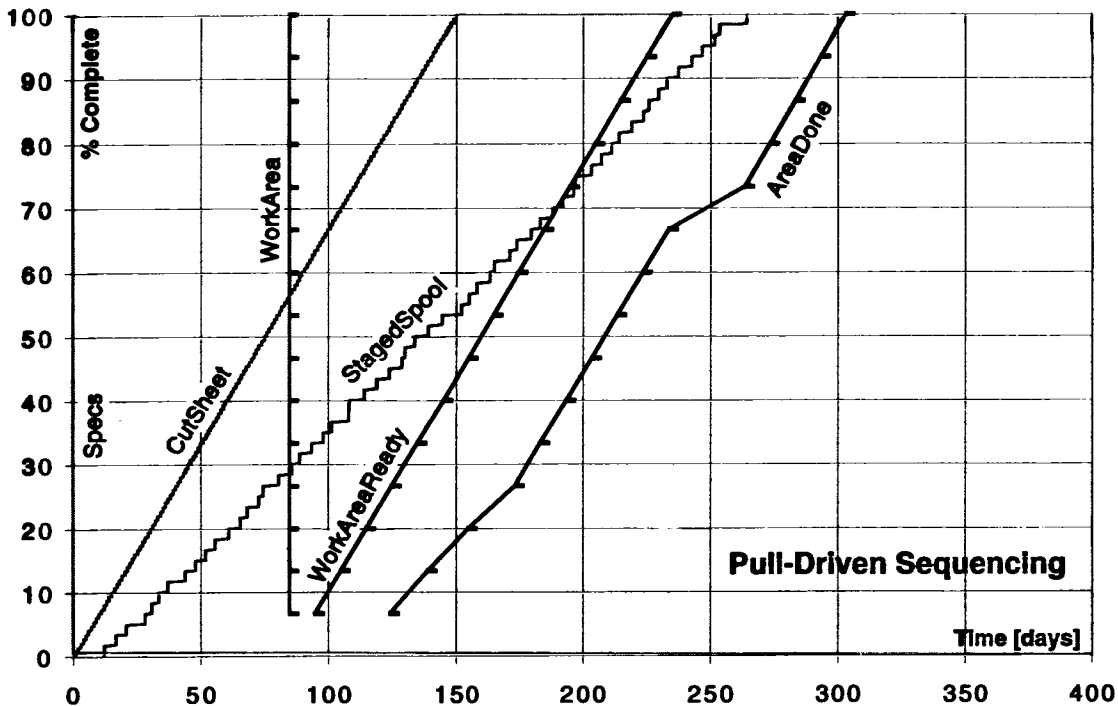


Figure 14: Single Run Percent Complete vs. Time for Pull-driven Sequencing

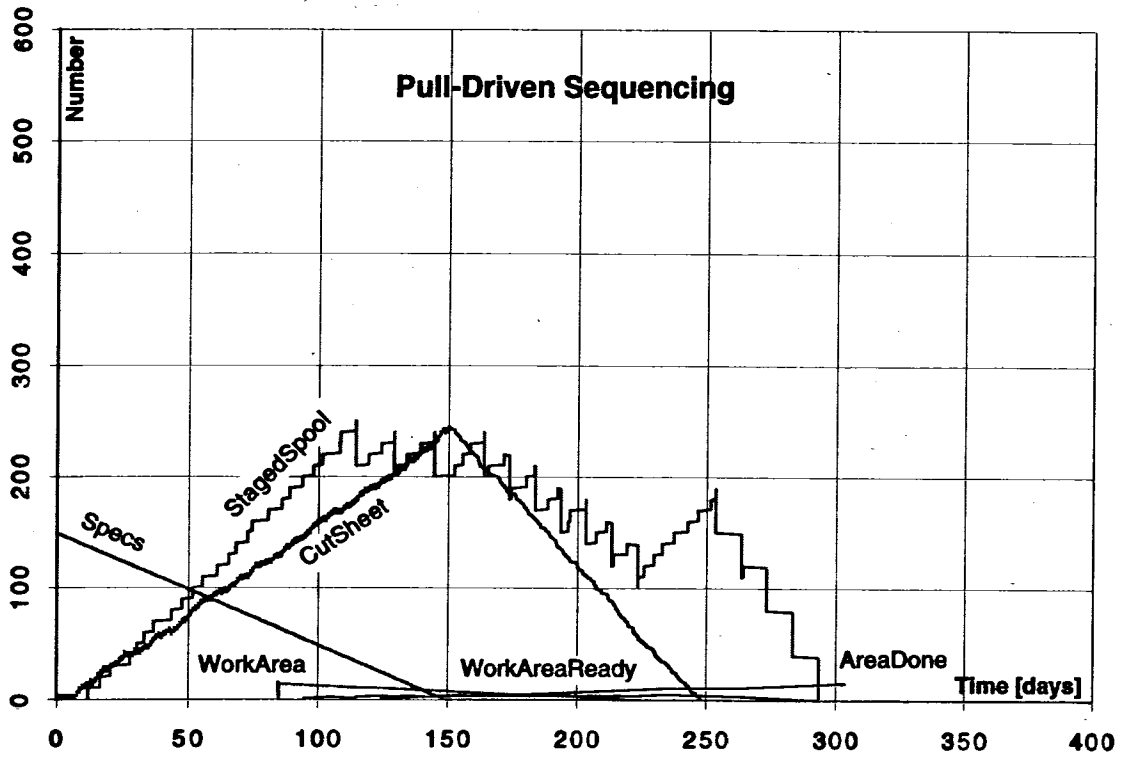


Figure 15: Single Run Number Resources vs. Time for Pull-driven Sequencing

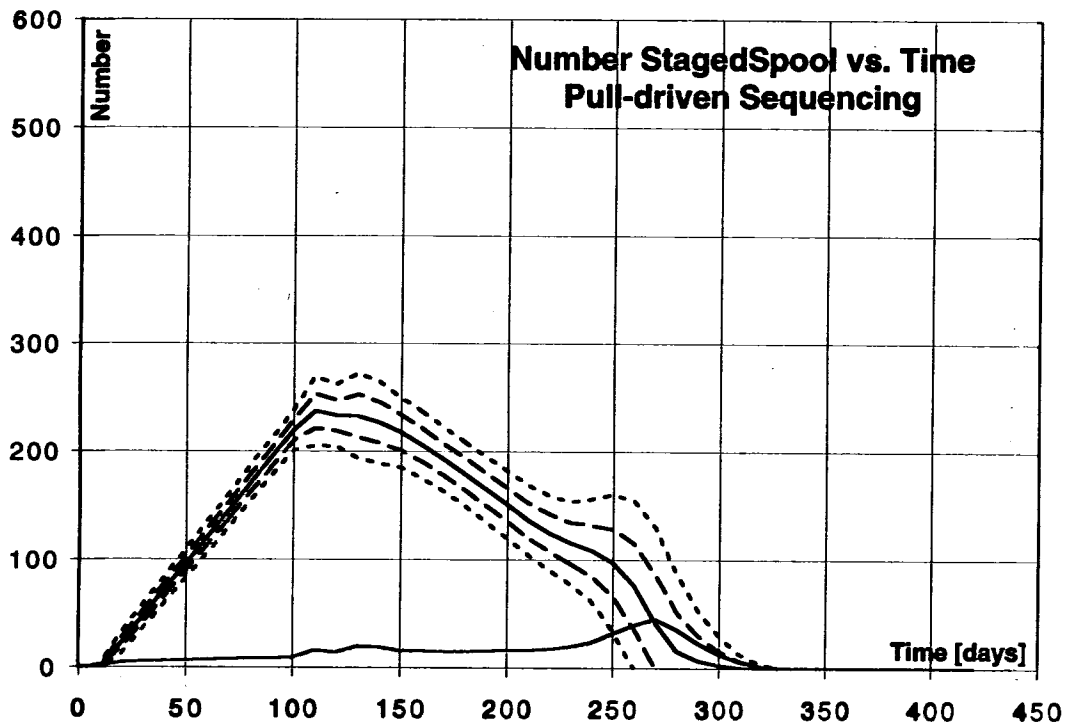


Figure 16: 1000 Iterations Number StagedSpool vs. Time for Pull-driven Sequencing

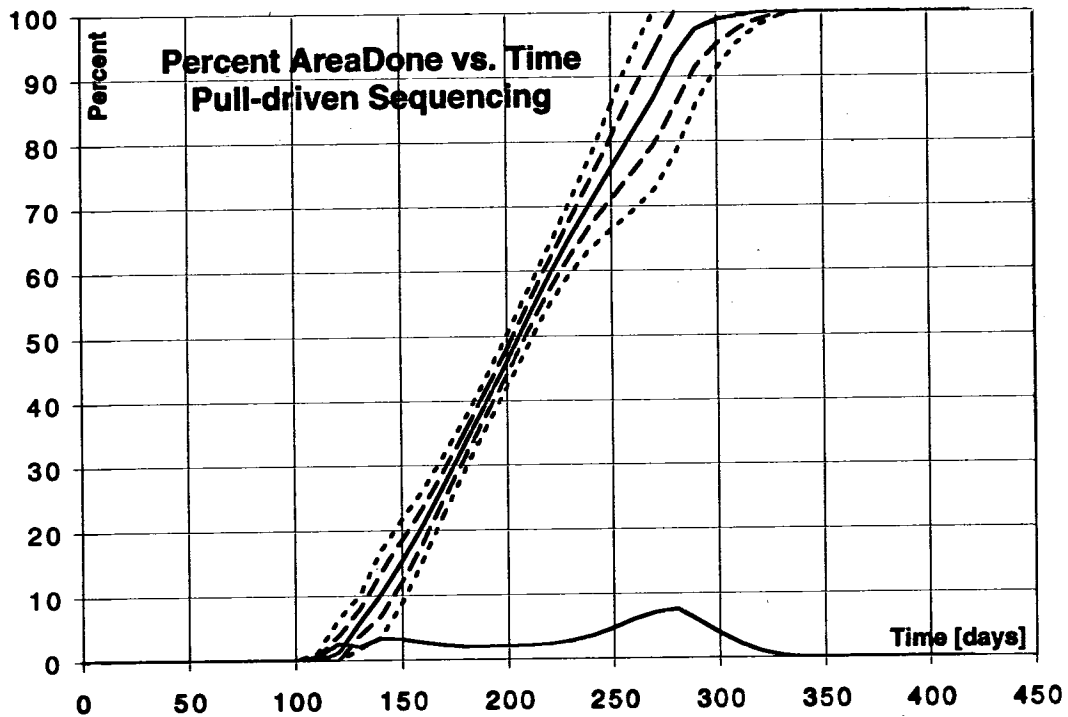


Figure 17: 1000 Iterations Percent Area Done vs. Time for Pull-driven Sequencing

8. IMPLEMENTATION HARDWARE AND SOFTWARE

All models were run in STROBOSCOPE (version 1, 2, 2, 0) on a Pentium 200-Mhz computer running Windows® 95. Source code is included in Appendix 2 of this report so that the reader, interested in further experimenting with alternative inputs to this model, can do so.

9. CONCLUSIONS

This report has shown how the lean-production "pull" technique can be applied to improve the execution of a construction process. Fast-track construction projects (projects in which on-site work starts prior to completion of design or fabrication of all materials) that require assembly of unique parts and that are plagued by many uncertainties are difficult to schedule accurately and in detail in advance. As uncertainties manifest themselves during project execution, the production schedule must be adjusted in a flexible manner for field work to progress efficiently and for work-in-progress inventories to remain small. The pull technique suggests that real-time feedback from construction be used to drive the sequencing of off-site work, and vice versa. By choosing to process "matching parts" first, the downstream process will be able to proceed in a more expedient fashion, and completed units will be available sooner than would be the case otherwise.

Only one pull link was implemented in the model discussed here. Obviously, choosing where, when, and how to pull is an important issue. Many pull links could be created, but each costs money to implement and the effects of one may offset those of another. Investigation of this issue must be supported by collection of process data that describes activities, resources, and duration and path-flow uncertainties of the system that is to be improved. Discrete-event simulation can help the decision-maker understand the system's behavior and gauge the impact pull links may have. Using the simulated data, a cost-benefit analysis can then be performed prior to physically establishing those links.

The collection of process data in and by itself is a worthwhile endeavor. Knowing where uncertainties exist and how large they are will help focus on reducing those uncertainties (Ballard and Howell 1997). It should be obvious from the limited work that has been conducted to date on implementing production control in construction as is advocated by the lean production philosophy, that process-level analysis and control of construction is promising area of research, development, and application.

10. ACKNOWLEDGMENTS

I am most indebted to Mr. James C. Goodwin, Manager of Materials Management at H.B. Zachry, for letting me study industry practices at the Lyondell-Citgo Refinery Expansion Project site in Houston, TX. This study increased my awareness and understanding of the complexity of construction materials management.

Thanks are due to Professor Julio C. Martinez, at the Virginia Polytechnic and State University, for making STROBOSCOPE readily available. Thanks to him as well as his thesis advisor, Professor Photios G. Ioannou at The University of Michigan, this expressive and especially fast discrete-event simulation system now exists to efficiently model construction processes.

Last, but not least, I owe thanks to my Berkeley colleague, Mr. H. Glenn Ballard, for introducing me to the "Last Planner" and discussing field implementations of lean construction, which has sharpened my own thinking about this subject. Glenn also provided valuable feedback on drafts of this report.

This research was funded by grant CMS-9622308 from the National Science Foundation, whose support is gratefully acknowledged. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the author and do not necessarily reflect the views of the National Science Foundation.

11. APPENDIX 1: REFERENCES

- Alarcon, L. (ed.)(1997). *Lean Construction*. Balkema, Rotterdam, The Netherlands, 497 pp.
- Ballard, G. and Howell, G. (1997). "Shielding Production from Uncertainty: First Step in an Improvement Strategy." *Technical Report 97-1*, Construction Engineering and Management Program, Civil & Environmental Engineering Department, University of California, Berkeley, CA.
- Bernold, L.E. and Salim, Md. (1993). "Placement-oriented design and delivery of concrete reinforcement." *Journal of Construction Engineering and Management*, ASCE,

- New York, NY, 119(2), 323-335.
- BRT (1983). *More Construction for the Money. Summary Report Construction Industry Cost Effectiveness Project*. Business Roundtable, 96 pp. New York, NY.
- Chamberlain, W.W. (1984). "KANBAN or MRP: What is the Real Thing?" in *Readings in Management and Personal Development, APICS 27th Annual Intl. Conf.*, 9-12 October 1984, Las Vegas, NV, pp. 6-10.
- CII (1987). *Project Materials Management Handbook*. Materials Management Task Force, CII, Univ. of Texas, Austin, TX.
- Halpin, D.W. and Riggs, L.S. (1992). *Planning and Analysis of Construction Operations*. Wiley-Interscience, New York, NY, 381 pp.
- Howell, G.A. and Ballard, H.G. (1995). *Managing Uncertainty in the Piping Function*. Report to the Construction Industry Institute, The University of Texas at Austin, TX, November, 103 pp.
- Howell, G., Laufer, A., and Ballard, G. (1993). "Interaction between sub-cycles: one key to improved methods." *Journal of Construction Engineering and Management*, ASCE, New York, NY, 119(4), 714-728.
- Koskela, L. (1992). *Application of the New Production Philosophy to Construction*. Technical Report 72, CIFE, Stanford University, Stanford, CA, September, 75 pp.
- Martinez, J.C. (1996). *STROBOSCOPE State and Resource Based Simulation of Construction Processes*. Ph.D. Dissertation, Civil & Envir. Engrg. Dept., Univ. of Michigan, Ann Arbor, MI, 518 pp.
- Melles, B. and Wamelink, J.W.F. (1993). *Production Control in Construction*. Delft University Press, Delft, The Netherlands, 320 pp.
- O'Brien, W.J. (1995). "Construction Supply Chains: Case Study and Integrated Cost and Performance Analysis." *Proceedings IGLC-3*, held in Albuquerque, NM, pp. 187-222 in Alarcon (1997).
- Odeh, A.M. (1992). *CIPROS: Knowledge-based Construction Integrated Project and Process Planning Simulation System*. Ph.D. Dissertation, Civil & Envir. Engrg. Dept., Univ. of Michigan, Ann Arbor, MI.
- Oglesby, C., Parker, H., and Howell, G. (1989). *Productivity Improvement in Construction*. McGraw-Hill, Inc., 588 pp.
- Perry, C. and Grieg, I.D. (1975) "Estimating the mean and variance of subjective distributions in PERT and Decision analysis." *Management Science*, 21 (12) 1477-1480.
- Phair, M., Tulacz, G.L., and Angelo, W.J. (1997). "Cushier Controls, Cabs, Make Cash: Ergonomics and Comfort Go Hand-in-Hand with Productivity." *ENR*, Feb. 10, pp. 34-39.
- Tatum, C.B. (1985). "Evaluating Construction Progress." *Project Management Journal*, Special Summer Issue, August, 52-57.
- Thabet, W.Y. and Beliveau, Y.J. (1993). "A Model to Quantify Work Space Availability for Space Constrained Scheduling within a CAD Environment." *Proc. 5th Intl. Conf. Comp. Civil and Bldg. Engrg.*, 110-116, ASCE, NY, NY.
- Thomas, H.R., Sanvido, V.E., and Sanders, S.R. (1989). "Impact of Material Management on Productivity—a Case Study." *Journal of Construction Engineering and Management*, ASCE, New York, NY, 115 (3) 370-384.
- Tommelein, I.D. (1997a). "Discrete-event Simulation of Lean Construction Processes." *Proc. Conf. of the International Group for Lean Construction (IGLC-5)*, 16-17 July 1997 at

Griffith University-Gold Coast Campus, Gold Coast, Queensland, Australia, in press.

Tommelein, Iris D. (1997b). "Models Of Lean Construction Processes: Example Of Pipe-Spool Materials Management." *Proc. Construction Congress '97*, Oct. 5-7 in Minneapolis, MN, ASCE, New York, NY, in press.

Tommelein, I.D., Carr, R.I., and Odeh, A.M. (1994). "Assembly of Simulation Networks using Designs, Plans, and Methods." *Journal of Construction Engineering and Management*, ASCE, New York, NY, 120(4), 796-815.

Womack, J.P. and Jones, D.T. (1996). *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. Simon & Schuster, New York, NY, 350 pp.

Womack, J.P., Jones, D.T., and Roos, D. (1990). *The Machine that Changed the World*. New York: Harper Perennial.

12. APPENDIX 2: SOURCE CODE FOR PIPE-SPOOL MODEL

```

/*****
/ PULL-DRIVEN SCHEDULING
/ 31 May 97 IDT
/
/ The assumptions made in this model are described in the report:
/
/ Tommelein, Iris D. (1997). Discrete-Event Simulation of a
/ Materials-Handling Process that Requires Resource Matching:
/ Example of Pipe-Spool Installation. Technical Report 97-2,
/ Construction Engineering and Management Program, Department
/ of Civil and Environmental Engineering, University of
/ California, Berkeley, CA.
/
/ In this model, 600 pipe spools are to be installed in 15 areas,
/ so there are 40 spools per area.
/
/ Time units are [days].
/
/*****
/* INPUT PARAMETERS

OUTFILE OutputFile "PipeSpool.sto";

VARIABLE SingleRun 0;
/ Set this variable to 1 meaning yes (0 = no) when you want to collect
/ data on elements in queues for a SINGLE simulation run, and print
/ that data to the OutputFile file. This file can then be used to
/ generate plots using Excel.
/ When SingleRun equals 0, MULTIPLE runs will be performed in order
/ to collect statistical data.

/ Print the column headers for the spreadsheet table:
IF SingleRun;
    PRINT OutputFile
        "\t Queue \t SimTime \t StagedSpool \t WorkAreaReady \t AreaDone
\t DefectSpool \t CutSheet \t WorkArea \t Specs \t Feedback \t Status \n";
ENDIF;

/*****
/* Definition of resource types

/=== OFF SITE ===

GENTYPE AStart;

CHARTYPE ATextFile;
SUBTYPE ATextFile ASpec;
SAVEPROPS ATextFile SpecNo RandomNo;
/ SpecNo is the number of the specification (Spec), where each Spec will
/ refer to exactly one isometric (ISO) drawing.
/ RandomNo truly is a random number. This will be used to define the
/ draw order to model the uncoordinated and the pull-driven processes.
```

```

CHARTYPE AGraphicFile;
SUBTYPE AGraphicFile ACutSheet;
SAVEPROPS AGraphicFile SpecNo DwgNo RandomNo FinalAreaNo AreaReady;
/ SpecNo is the number of the Spec used as the basis from which to
/   create cutsheets. It is assumed (later in this file) that there
/   are 4 cutsheets per ISO.
/ DwgNo is the number of the cutsheet drawing, and each cutsheet will
/   refer to exactly one pipe spool's TagNo.
/ RandomNo truly is a random number. This will be used to define the
/   draw order to model the uncoordinated and the pull-driven processes.
/ FinalAreaNo records which area the spool must end up in.
/ AreaReady is for the pull-driven model to re-sequence spools for
/   fabrication to match progress on site.

GENTYPE ADesignTeam;

CHARTYPE AMaterial;
SUBTYPE AMaterial ASpool;
SAVEPROPS AMaterial TagNo RandomNo FinalAreaNo AreaReady;
/ TagNo is the spool's identification number that corresponds to the
/   cutsheet's drawing number (DwgNo).

GENTYPE AFabCrew;

/=== ON SITE ===

CHARTYPE ASpace;
SUBTYPE ASpace AnArea;
SAVEPROPS ASpace AreaNo;
/ AreaNo is a specific number assigned to each area.
/ Spaces need not have a RandomNo property, they need not be in random
/ order. In any model, any order will appear to be random during
/ simulation, compared to the random ordering of spools.

GENTYPE APrepCrew;

/ Resource to Provide Feedback Necessary to Pull

CHARTYPE ADataPiece;
SUBTYPE ADataPiece ADatum;
SAVEPROPS ADataPiece AreaReadyNo;

/ === JOINED ===

GENTYPE AnInstallCrew;

COMPTYPE AnAreaDone;
SAVEPROPS AnAreaDone MinValTagNo MaxValTagNo AreaNo;

/*****
/* Definition of network nodes

```

```

/=== OFF SITE ===

QUEUE StartOffSite AStart;
COMBI OffSiteWork;
QUEUE Specs ATextFile;
QUEUE DesignTeam ADesignTeam;
COMBI Design;
QUEUE CutSheet AGraphicFile;
QUEUE FabCrew AFabCrew;
COMBI Fabricate;
FORK GoodBad AMaterial;
QUEUE DefectSpool AMaterial;
/ Rework must take place before Fabricate so Rework gets a higher
/ PRIORITY (default = 0) [Martinez 1996 p. 98]
COMBI Rework;
    PRIORITY Rework 1;
CONSOLIDATOR AwaitTransport;
NORMAL Transport;

/=== ON SITE ===

QUEUE StartOnSite AStart;
COMBI FieldWork;
QUEUE WorkArea ASpace;
QUEUE PrepCrew APrepCrew;
COMBI PrereqWork;

/=== JOINED ===

QUEUE StagedSpool AMaterial;
QUEUE WorkAreaReady ASpace;
QUEUE InstallCrew AnInstallCrew;
COMBI Install;
ASSEMBLER SpoolInArea AnAreaDone;
QUEUE AreaDone AnAreaDone;

/=== PULL ===

QUEUE Feedback ADataPiece;
/ Update must take place before Fabricate so Update gets a higher
/ PRIORITY (default = 0) [Martinez 1996 p. 98]
COMBI Update;
    PRIORITY Update 1;

/*****
/* Definition of network Links

/=== ON SITE ===
/ Specify on site before off site this time, so link priorities will be
/ correct.

LINK ON1 StartOnSite FieldWork;
LINK CR5 PrepCrew PrereqWork;
LINK CR6 PrereqWork PrepCrew;
LINK WA1 FieldWork WorkArea;
LINK WA2 WorkArea PrereqWork;
LINK WA3 PrereqWork WorkAreaReady;

```



```

LINK WA4 WorkAreaReady Install;
/ Define WA4 before PS8 (and CR7) to make sure work area will be drawn
/ before the spool or the crew, as set of spools drawn must match the
/ work area. [Martinez 1996 p. 89]

/=== OFF SITE ===

LINK OF1 StartOffSite OffSiteWork;
LINK SP1 OffSiteWork Specs;
LINK SP2 Specs Design;
LINK DT1 DesignTeam Design;
LINK DT2 Design DesignTeam;
LINK DW1 Design CutSheet;
LINK DW2 CutSheet Fabricate;
LINK CR1 FabCrew Rework;
LINK CR2 Rework FabCrew;
LINK CR3 FabCrew Fabricate;
LINK CR4 Fabricate FabCrew;
LINK PS1 Fabricate GoodBad;
LINK PS2 GoodBad AwaitTransport;
LINK PS3 GoodBad DefectSpool;
LINK PS4 DefectSpool Rework;
LINK PS5 Rework AwaitTransport AMaterial;
LINK PS6 AwaitTransport Transport AMaterial;
LINK PS7 Transport StagedSpool;
LINK PS8 StagedSpool Install;

/=== JOINED ===

LINK CR7 InstallCrew Install;
LINK CR8 Install InstallCrew;
LINK WA5 Install SpoolInArea ASpace;
LINK PS9 Install SpoolInArea AMaterial;
LINK AD1 SpoolInArea AreaDone;

/=== Pull ===

LINK FB1 PrereqWork Feedback;
LINK DW3 CutSheet Update;
LINK FB2 Feedback Update;
/ Define DW3 before FB2 so CutSheet gets drawn before Feedback as there
/ will be many pieces of feedback data that are essentially the same.
LINK DW4 Update CutSheet;

/+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
/ OFF-SITE ACTIVITY SEQUENCE
/+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

/*****
/* Statements to assist in the definition of attributes of StartOffSite
/ and its related links

INIT StartOffSite 1;

/*****
/* Startup of OffSiteWork
```

```

DURATION OffSiteWork '0';

/*****
/* Termination of OffSiteWork

VARIABLE TotNoSpecs 150;

BEFOREEND OffSiteWork GENERATE TotNoSpecs ASpec;
/ Number each specification in the order in which it is created.
/ Since all specs are generated at the same simulation time and none will
/ have had the chance to leave the Specs queue, Count will be able to
/ uniquely number each resource entering the Specs queue.
ONRELEASE SP1 ASSIGN SpecNo Specs.ASpec.Count+1; / [Martinez 1996 p. 204]
/ Then assign a random number so their draw sequence (or a discipline
/ that sorts them upon entry into a queue) can be "random".
ONRELEASE SP1 ASSIGN RandomNo Rnd[];

/*****
/* Entry of resources into Specs

SAVEVALUE EnteringSpec 0;
ONENTRY Specs ASSIGN EnteringSpec EnteringSpec+1;

IF SingleRun;
  ONENTRY Specs PRINT OutputFile
    "\t Specs \t %.2f \t\t\t\t\t\t\t %.2f \t\t entered \n"
    SimTime 100*EnteringSpec/TotNoSpecs;
  ONENTRY Specs PRINT OutputFile
    "\t Specs \t %.2f \t\t\t\t\t\t\t %f \t\t in \n"
    SimTime Specs.ASpec.Count;
ENDIF;

/*****
/* Statements to assist in the definition of attributes of DesignTeam
/ and its related links

INIT DesignTeam 1;

/*****
/* Startup of Design

IF SingleRun;
  ONDRAW SP2 PRINT OutputFile
    "\t Specs \t %.2f \t\t\t\t\t\t\t %.0f \t\t out \n"
    SimTime Specs.CurCount;
ENDIF;

DURATION Design '1';
/ It takes about 1 day [8 hrs] to create an ISO.
/ At this time, I have no data on how much time it takes to create
/ cutsheets once the ISO has been completed.

/*****
/* Termination of Design

BEFOREEND Design GENERATE 4 ACutSheet;
/ On average there will be 4 cutsheets per ISO [cf. Mike Villejas at

```

```

/ Jacobs Engineering, personal communication (February 1997)].

SAVEVALUE TempSpecNo 0;
BEFOREEND Design ASSIGN TempSpecNo Design.ATextFile.SpecNo;

ONRELEASE DW1 ASSIGN SpecNo TempSpecNo;
ONRELEASE DW1 ASSIGN RandomNo Rnd[];

/ Number the cutsheets incrementally.
SAVEVALUE NextCutSheetNo 0;
ONRELEASE DW1 ASSIGN NextCutSheetNo NextCutSheetNo+1;
ONRELEASE DW1 ASSIGN DwgNo NextCutSheetNo;

/ Next, figure out which final area the spool will have to be installed in.
/ This is computed here, though in reality it reflects information shown
/ in specs and on design dwgs. The formula results in spools 1 through
/ 40 going to area 1001, 41 through 80 to area 1002, etc.; that is: 40
/ spools per area.
ONRELEASE DW1 ASSIGN FinalAreaNo 1001+Int[(DwgNo-1)/40];
ONRELEASE DW1 ASSIGN AreaReady 0;
/ 0 means area on site is NOT yet ready for spool installation, so
/ that the spool's fabrication is not "urgent".

/*****
/* Entry of resources into CutSheet

SAVEVALUE EnteringCutSheet 0;
ONRELEASE DW1 ASSIGN EnteringCutSheet EnteringCutSheet+1;
/ Note that there are two links going into the CutSheet Queue.
/ The "entering" cutsheet counter should ONLY count cutsheets being
/ output by the Design combi activity and NOT count the updates
/ being done through link DW3.

IF SingleRun;
  ONENTRY CutSheet PRINT OutputFile
    "\t CutSheet \t %.2f \t\t\t\t\t %.2f \t\t\t\t\t entered \n"
    SimTime 100*EnteringCutSheet/(TotNoSpecs*4);
    / Compute % of cutsheets that have been prepared.
  ONENTRY CutSheet PRINT OutputFile
    "\t CutSheet \t %.2f \t\t\t\t\t %.0f \t\t\t\t\t in \n"
    SimTime CutSheet.ACutSheet.Count;
ENDIF;

/*****
/* Statements to assist in the definition of attributes of FabCrew and
/* its related links

INIT FabCrew 20;

/*****
/* Startup of Fabricate

```

```

/ Coordinated Model:
/   Process cutsheets in numerical order.
/   This will be the case by default. No DRAWORDER is to be specified.
/ Pull and Random Model:
/   First process CutSheets for which matching area is ready;
/   then proceed with random ordering.
DRAWORDER DW2 '-AreaReady-RandomNo';
/ This draw order reflects that spools for which the corresponding
/ installation area is ready on site (AreaReady = 1) must get fabricated
/ ahead of others and then rushed to the site. This expression works
/ because by construction:
/   AreaReady takes on a value of 0 or 1;
/   SpecNo, DwgNo, TagNo, or FinalAreaNo takes on a value in thousands.
/   Random number ranges from 0 to 1 (0 <= Rnd[] <= 1).
/ and DRAWORDER takes smallest numbers first (ascending order).
/ In case of the Random model, using DRAWORDER DW2 'RandomNo'; would
/ work equally well as all CutSheets have the value AreaReady == 0
/ by default. Note however that the numerical results will be different
/ when the simulation is run with a specified seed because of the minus
/ sign in front of the random number.

IF SingleRun;
  ONDRAW DW2 PRINT OutputFile
    "\t CutSheet \t %.2f \t\t\t\t\t %.0f \t\t\t\t\t out \n"
    SimTime CutSheet.CurCount;
ENDIF;

/ Deterministic Model:
/ Use a constant value of 5, which is the 'most likely' value
/ from the Pertpg distribution.
/DURATION Fabricate 5;
/ Random, Coordinated, and Pull-driven Model:
/ Use the probabilistic duration value from CII report by Howell
/ and Ballard (1995 p. 19).
DURATION Fabricate Pertpg[3,5,14];

/*****
/* Termination of Fabricate
/ For each cutsheet fabricate one spool.

BEFOREEND Fabricate GENERATE 1 ASpool;
/ Each spool gets the number of the drawing (cutsheet) that shows
/ its design.
BEFOREEND Fabricate
  ASSIGN Fabricate.AMaterial.TagNo Fabricate.AGraphicFile.DwgNo;
/ Other property values can also be passed along for checking purposes,
/ though they are not needed further downstream.
BEFOREEND Fabricate
  ASSIGN Fabricate.AMaterial.RandomNo
    Fabricate.AGraphicFile.RandomNo;
BEFOREEND Fabricate
  ASSIGN Fabricate.AMaterial.FinalAreaNo
    Fabricate.AGraphicFile.FinalAreaNo;
BEFOREEND Fabricate
  ASSIGN Fabricate.AMaterial.AreaReady
    Fabricate.AGraphicFile.AreaReady;

```

```

/*****
/* Activation of successors and routing of resources through GoodBad

/ Deterministic Model:
/ Assume there is no rework so spools will never go to DefectSpool.
/STRENGTH      PS2 '100';
/STRENGTH      PS3 '0';
/ Random, Coordinated, and Pull-driven Models:
/ Assume that 10% of the spools require rework [cf. Glenn Ballard
/ (1996), personal communication: "industry spool defect rate
/ ranges from 1 to 10%"].
STRENGTH      PS2 '90';
STRENGTH      PS3 '10';

/*****
/* Entry of resources into DefectSpool

SAVEVALUE EnteringDefectSpool 0;
ONENTRY DefectSpool ASSIGN EnteringDefectSpool EnteringDefectSpool+1;

IF SingleRun;
  ONENTRY DefectSpool PRINT OutputFile
    "\t DefectSpool \t %.2f \t\t\t\t %.2f \t\t\t\t\t entered \n"
    SimTime 100*EnteringDefectSpool/(TotNoSpecs*4);
    / Compute % of spools that are defective.
  ONENTRY DefectSpool PRINT OutputFile
    "\t DefectSpool \t %.2f \t\t\t\t\t %.0f \t\t\t\t\t in \n"
    SimTime DefectSpool.ASpool.Count;
ENDIF;

/*****
/* Startup of Rework

IF SingleRun;
  ONDRAW PS4 PRINT OutputFile
    "\t DefectSpool \t %.2f \t\t\t\t\t %.0f \t\t\t\t\t out \n"
    SimTime DefectSpool.CurCount;
ENDIF;

DURATION Rework Pertpg[3,5,14];
/ Assume a pretty bad situation (though not necessarily the worst
/ situation possible): rework takes the same amount of time as
/ fabricating a new spool.

/*****
/* Termination of AwaitTransport

CONSOLIDATEWHEN AwaitTransport 'AwaitTransport.ASpool.Count>=10';
/ Consolidate 10 spools for each shipment.

/*****
/* Startup of Transport

/ Deterministic Model:
/ Constant value of 3.
/DURATION Transport 3;
/ Random, Coordinated, and Pull-driven Models:

```

```

/ IDT assumes 3 days plus or minus 1 day.
DURATION Transport Normal[3,1];

/*****
/* Entry of resources into StagedSpool

SAVEVALUE EnteringStagedSpool 0;
ONENTRY StagedSpool ASSIGN EnteringStagedSpool EnteringStagedSpool+1;

IF SingleRun;
  ONENTRY StagedSpool PRINT OutputFile
    "\t StagedSpool \t %.2f \t %.2f \t\t\t\t\t\t\t entered \n"
    SimTime 100*EnteringStagedSpool/(TotNoSpecs*4);
    / Keep track of the number of spools that entered this queue
    / and compute it as a % of the total number of spools.
  ONENTRY StagedSpool PRINT OutputFile
    "\t StagedSpool \t %.2f \t %.0f \t\t\t\t\t\t\t in \n"
    SimTime StagedSpool.ASpool.Count;
ENDIF;

/+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
/ ON-SITE ACTIVITY SEQUENCE
/+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

/*****
/* Statements to assist in the definition of attributes of StartOnSite
/ and its related links

INIT StartOnSite 1;

/*****
/* Startup of FieldWork

DURATION FieldWork '85';
/ This number has been chosen by IDT to reflect build-up of buffer of
/ pipe spools for 95 days (WorkAreas available at time 85 and the first
/ one readied 10 days later) before installation starts (see deterministic
/ model for first WorkAreaReady at time 95).
/ Practitioners say they like spools to be on site for 5 to 6 months
/ prior to installation! Counting 20 working days per month, 95 days
/ is nearly 5 months. Of course, the larger the buffer, the smaller the
/ impact of uncertainty on the downstream process.

/*****
/* Termination of FieldWork

VARIABLE TotNoAreas 15;

BEFOREEND FieldWork GENERATE TotNoAreas AnArea;

ONRELEASE WA1 ASSIGN AreaNo WorkArea.AnArea.Count+1001;
/ This numbers each area in the order in which it is created.
/ The number is expressed in thousands (4 digits) as opposed to being a
/ single- or double-digit number. This was determined rather arbitrarily;
/ otherwise there are just too many simple 1s and 2s to keep track of in
/ this model. However,note that DRAWORDER DW2 relies on this convention.

```

```

/*****
/* Entry of resources into WorkArea

DISCIPLINE WorkArea 'AreaNo';

SAVEVALUE EnteringWorkArea 0;
ONENTRY WorkArea ASSIGN EnteringWorkArea EnteringWorkArea+1;

IF SingleRun;
  ONENTRY WorkArea PRINT OutputFile
    "\t WorkArea \t %.2f \t\t\t\t\t %.2f \t\t\t entered \n"
    SimTime 100*EnteringWorkArea/TotNoAreas;
    / Keep track of the number of areas that entered this queue
    / and compute it as a % of the total number of areas.
  ONENTRY WorkArea PRINT OutputFile
    "\t WorkArea \t %.2f \t\t\t\t\t %.0f \t\t\t in \n"
    SimTime WorkArea.AnArea.Count;
ENDIF;

/*****
/* Statements to assist in the definition of attributes of PrepCrew and
/ its related links

INIT PrepCrew 1;

/*****
/* Startup of PrereqWork

IF SingleRun;
  ONDRAW WA2 PRINT OutputFile
    "\t WorkArea \t %.2f \t\t\t\t\t %.0f \t\t\t out \n"
    SimTime WorkArea.CurCount;
ENDIF;

DURATION PrereqWork '10';

/*****
/* Termination of PrereqWork

/-----
/ Feedback mechanism for pulling
/-----
/ The dummy CurrentAreaNo keeps track of the current area number so that
/ each piece of feedback generated in the PreReqWork combi gets this
/ number associated with it. It is needed because there are 40 such
/ pieces generated for each combi instance and ASSIGN only works on
/ one element at a time. Thus, ONRELEASE is used (instead of ASSIGN)
/ to assign the area number to the piece of feedback. At the moment
/ ONRELEASE gets evaluated, though, the combi instance has been
/ terminated and so the ASpace resource (needed to evaluate
/ PrereqWork.ASpace.AreaNo) has been destroyed, whereas the dummy can
/ explicitly keep track of this value, beyond the life of the combi
/ instance.
/ Note that STROBOSCOPE will not confuse values of CurrentAreaNo when
/ multiple instances of PreReqWork exist concurrently as the assignment
/ is instance-specific.

```

```

SAVEVALUE Pulling 1;
IF Pulling;

SAVEVALUE CurrentAreaNo 0;
BEFOREEND PrereqWork GENERATE 40 ADatum;
BEFOREEND PrereqWork ASSIGN CurrentAreaNo PrereqWork.ASpace.AreaNo;
ONRELEASE FB1 ASSIGN AreaReadyNo CurrentAreaNo;

/*****
/* Startup of Update
/-----
/ Matching Feedback with CutSheets
/-----
/ For other examples using FILTERS see Martinez (1996) Chapter 12.
/ SYNTAX: FILTER FilterName CharType CursoredFilterExpression;
/ SYNTAX: VARPROP TypeName VarPropName ExpressionINTermsOfOtherProps;
/-----
/Dummy filter to be refined later; any datapiece will do for now.
FILTER DataThatMatches ADataPiece 1;

/ First make sure the cutsheet being considered has not been updated yet.
/ Next there must be at least one piece of feedback in the Feedback queue
/ or otherwise there is nothing to be checked. As there may be several
/ (up to 40) data pieces in that queue that match, take the MinVal to get
/ a single value (they will all have the same value anyway).
VARPROP AGraphicFile FeedBackAvailable
  'AreaReady==0 &
  Feedback.DataThatMatches.Count>0 &
  Feedback.DataThatMatches.AreaReadyNo.MinVal==FinalAreaNo';

/ A cutsheet can be updated if matching feedback is available:
FILTER DocumentCanBeUpdated AGraphicFile FeedBackAvailable;

/ Find which specific cutsheet is being considered for updating:
VARIABLE CurrentlyCursoredAreaNo
  'DocumentCanBeUpdated.HasCursor
  ? DocumentCanBeUpdated.FinalAreaNo
  : DW3.FinalAreaNo';

/ For that specific cutsheet, find the matching feedback. Feedback
/ matches when the area that is ready (AreaReadyNo) equals the final
/ installation area of the spool shown on the cutsheet (FinalAreaNo).
FILTEREXP DataThatMatches 'AreaReadyNo==CurrentlyCursoredAreaNo';

ENOUGH DW3 CutSheet.DocumentCanBeUpdated.Count; / 1 is enough
DRAWWHERE DW3 FeedBackAvailable;
DRAWWHERE FB2 AreaReadyNo==Update.AGraphicFile.FinalAreaNo;

/*****
SAVEVALUE EnteringFeedback 0;
ONENTRY Feedback ASSIGN EnteringFeedback EnteringFeedback+1;

```



```

IF SingleRun;
  ONENTRY Feedback PRINT OutputFile
    "\t Feedback \t %.2f \t\t\t\t\t\t\t %.2f \t entered \n"
    SimTime 100*EnteringFeedback/(TotNoAreas*40);
    / Keep track of the number of datapieces that entered this queue
    / and compute it as a % of the total number of data pieces.
  ONENTRY Feedback PRINT OutputFile
    "\t Feedback \t %.2f \t\t\t\t\t\t\t %.0f \t in \n"
    SimTime Feedback.ADatum.Count;
ENDIF;

IF SingleRun;
  ONDRAW FB2 PRINT OutputFile
    "\t Feedback \t %.2f \t\t\t\t\t\t\t %.0f \t out \n"
    SimTime Feedback.CurCount;
ENDIF;

/*****
/* Termination of Update

ONRELEASE DW4 ASSIGN AreaReady 1;
IF SingleRun;
  ONRELEASE DW4 PRINT OutputFile
    "Update performed at time %.2f for CutSheet %.0f with FinalAreaNo
    %.0f generated from SpecNo %.0f \n"
    SimTime DwgNo FinalAreaNo SpecNo;
ENDIF;

ENDIF;
/ end of pulling mechanism
/-----

/*****
/* Entry of resources into WorkAreaReady

SAVEVALUE EnteringWorkAreaReady 0;
ONENTRY WorkAreaReady ASSIGN EnteringWorkAreaReady
    EnteringWorkAreaReady+1;

IF SingleRun;
  ONENTRY WorkAreaReady PRINT OutputFile
    "\t WorkAreaReady \t %.2f \t\t %.2f \t\t\t\t\t\t\t entered \n"
    SimTime 100*EnteringWorkAreaReady/TotNoAreas;
    / compute the % of total number of areas
  ONENTRY WorkAreaReady PRINT OutputFile
    "\t WorkAreaReady \t %.2f \t\t %.0f \t\t\t\t\t\t\t in \n"
    SimTime WorkAreaReady.AnArea.Count;
ENDIF;

/+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
/ COMBINED ACTIVITY SEQUENCE
/+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

/*****
/* Statements to assist in the definition of attributes of InstallCrew
/ and its related links

```

```

INIT InstallCrew 1;

/*****
/* Startup of Install

IF SingleRun;
  ONDRAW WA4 PRINT OutputFile
    "\t WorkAreaReady \t %.2f \t\t %.0f \t\t\t\t\t\t out \n"
    SimTime WorkAreaReady.CurCount;
  ONDRAW PS8 PRINT OutputFile
    "\t StagedSpool \t %.2f \t %.0f \t\t\t\t\t\t\t\t out \n"
    SimTime StagedSpool.CurCount;
ENDIF;

DURATION Install 10;

/-----
/ Strategy for matching spools with areas.
/-----
/ Areas are served in FIFO order if possible, otherwise serve the first
/ area for with the 40 matching spools are available. Each spool must
/ match the area that is drawn from the WorkAreaReady queue.
/-----
/ NOTE: need to define the WA4 link BEFORE the PS8 link,
/ i.e., first define the link through which areas are drawn.

/ SpoolThatMatches is a dummy filter for now.
/ Any spool will do; this filter will get refined later on.
FILTER SpoolThatMatches AMaterial 1;

/ VARPROP will be TRUE for Areas for which 40 matching spools are
/ available in the StagedSpool queue:
/ Count that there are at least 40 spools in the queue, if not, there is
/ no chance of finding 40 matching ones anyways. Count is also needed to
/ make sure at least one element is in the StagedSpool queue; otherwise,
/ STROBOSCOPE gives a run-time error.
VARPROP ASpace MatchingSpoolAvail
  'StagedSpool.SpoolThatMatches.Count>=40 &
  AreaNo==StagedSpool.SpoolThatMatches.FinalAreaNo.MinVal';

/ Filter to single out those areas for which a matching spool is available:
FILTER AreaWSpoolReady ASpace MatchingSpoolAvail;

/ Now get the specific area number of the area first in line for which a
/ matching spool is available:
VARIABLE NowCursoredAreaNo
  'AreaWSpoolReady.HasCursor ? AreaWSpoolReady.AreaNo : WA4.AreaNo';

/ Once we know which area we are talking about, get the corresponding
/ spool(s). This redefines the dummy filter that was used at first.
FILTEREXP SpoolThatMatches
  FinalAreaNo==NowCursoredAreaNo;

/ We have enough areas if there is at least one in the queue from which
/ we are going to draw for which the matching spool is available:
ENOUGH WA4 WorkAreaReady.AreaWSpoolReady.Count;

```

```

/ Now draw the area for which there is a set of matching spools:
DRAWWHERE WA4 MatchingSpoolAvail;

/ The attribute of PS8 assumes that the area is in the Install instance.
DRAWWHERE PS8 FinalAreaNo==Install.ASpace.AreaNo;

/ Note that in this model 40 spools will match, so DRAWUNTIL *is* needed,
/ otherwise only one spool will be drawn from the StagedSpool queue.
DRAWUNTIL PS8 Install.AMaterial.Count>=40;

/*****
/* Assembly of resources in SpoolInArea

ONASSEMBLY SpoolInArea
  ASSIGN AreaNo SpoolInArea.ASpace.AreaNo;

ONASSEMBLY SpoolInArea
  ASSIGN MinValTagNo SpoolInArea.AMaterial.TagNo.MinVal;

ONASSEMBLY SpoolInArea
  ASSIGN MaxValTagNo SpoolInArea.AMaterial.TagNo.MaxVal;

/*****
/* Entry of resources into AreaDone

SAVEVALUE EnteringAreaDone 0;
ONENTRY AreaDone ASSIGN EnteringAreaDone EnteringAreaDone+1;
FILTER AreaThatIsDone AnAreaDone 1;

IF SingleRun;
  ONENTRY AreaDone PRINT OutputFile
    "\t AreaDone \t %.2f \t\t\t %.2f \t\t\t\t\t entered \n"
    SimTime 100*EnteringAreaDone/TotNoAreas;
    / Compute % of areas completed.
  ONENTRY AreaDone PRINT OutputFile
    "\t AreaDone \t %.2f \t\t\t %.0f \t\t\t\t\t in \n"
    SimTime AreaDone.AreaThatIsDone.Count;
ENDIF;

/+-----+
/ Go Single Run!
/+-----+
/ Use the following two lines to run a single simulation.
/ The seed as shown is the one used in Tommelein's 1997 IGLC-5 and her
/ ASCE Construction Congress V papers.

IF SingleRun;
  SEED 527817728;
  SIMULATE;
ENDIF;

/+-----+
/ Go Multiple Runs!
/+-----+
/ When doing multiple runs, set the variable SingleRun at the beginning
/ of this file == 0. This will cause STROBOSCOPE to use the following
/ code to obtain certainty intervals on numbers of resources and times

```

```

/ to reach completion by collecting data over multiple simulation runs.

IF !SingleRun;

/*****
/* DATA COLLECTION
*****/
/ Save values of interest during each simulation run for each of the
/ following variables:
/   EnteringSpec <- not interesting; 150 Specs will be
/   available simultaneously at simulation time 0.
/   EnteringCutSheet
/   EnteringDefectSpool
/   EnteringStagedSpool
/   EnteringWorkArea <- not interesting; 15 Areas will be
/   available simultaneously at simulation time 85.
/   EnteringWorkAreaReady
/   EnteringAreaDone
/   EnteringFeedback
/ For the single-run case, this was done using SAVEVALUES in the
/ (see preceding code) but now multiple values need to be COLLECTed
/ and then statistically processed.

/ Save the project duration (which is when Install last finishes):
SAVEVALUE ProjectFinish 0;
ONENTRY AreaDone ASSIGN ProjectFinish
    'EnteringAreaDone==15&ProjectFinish==0 ? SimTime : 0';

/*****
/ Introduce the TimeStep combi so we will be certain that the
/ simulation clock reaches the exact times at which we want to
/ collect data.

VARIABLE TimeStepSize 10;

COMBI TimeStep;
SEMAPHORE TimeStep !TimeStep.CurInst;
/ Allow only one instance to exist at a time.
DURATION TimeStep TimeStepSize;
/ Data will be collected every X=TimeStepSize days.
BEFOREEND TimeStep PRINT OutputFile "Time is now %6.4f \n" SimTime;

/ Make sure the simulation does not go on forever (TimeStep would).
/ Set simulation end time at 450 which should be upper bound on the
/ project completion time using random sequencing.
VARIABLE EndTime 500;
SIMULATEUNTIL SimTime>=EndTime;

/*****
/* COLLECTORS
*****/
/ Define collectors that will gather data over multiple simulation runs.
/ Collectors need to be persistent (add '*' to their name to make them
/ persistent) or else the data will get lost when the CLEAR command is
/ invoked upon start of a new iteration.

```

```

COLLECTOR ProjDuration*;
COLLECTOR InstallFirstCollector*;

SAVEVALUE CurrentStep 0;

WHILE CurrentStep<=EndTime;

    COLLECTOR PercentCutSheetAtTime$<CurrentStep>$*;
    COLLECTOR NrCutSheetAtTime$<CurrentStep>$*;

    COLLECTOR PercentDefectSpoolAtTime$<CurrentStep>$*;
    COLLECTOR NrDefectSpoolAtTime$<CurrentStep>$*;

    COLLECTOR PercentStagedSpoolAtTime$<CurrentStep>$*;
    COLLECTOR NrStagedSpoolAtTime$<CurrentStep>$*;

    COLLECTOR PercentFeedbackAtTime$<CurrentStep>$*;
    COLLECTOR NrFeedbackAtTime$<CurrentStep>$*;

    COLLECTOR PercentWorkAreaReadyAtTime$<CurrentStep>$*;
    COLLECTOR NrWorkAreaReadyAtTime$<CurrentStep>$*;

    COLLECTOR PercentAreaDoneAtTime$<CurrentStep>$*;
    COLLECTOR NrAreaDoneAtTime$<CurrentStep>$*;

    ASSIGN CurrentStep CurrentStep+TimeStepSize;

WEND;

/*****
/* ITERATION
*****/

VARIABLE NrIterations 1000;
SAVEVALUE CurrentIteration* 1;
SAVEVALUE CurrentStepTwo* 0;

WHILE CurrentIteration<=NrIterations;

    CLEAR;

    INIT StartOffSite 1;
    INIT DesignTeam 1;
    INIT FabCrew 20;
    INIT StartOnSite 1;
    INIT PrepCrew 1;
    INIT InstallCrew 1;

    WHILE CurrentStepTwo<=EndTime;

        SIMULATEUNTIL SimTime>=CurrentStepTwo;

        COLLECT PercentCutSheetAtTime$<CurrentStepTwo>$
            100*EnteringCutSheet/(TotNoSpecs*4);
        COLLECT NrCutSheetAtTime$<CurrentStepTwo>$
            CutSheet.ACutSheet.Count;

```

```

COLLECT PercentDefectSpoolAtTime$<CurrentStepTwo>$
    100*EnteringDefectSpool/ (TotNoSpecs*4);
COLLECT NrDefectSpoolAtTime$<CurrentStepTwo>$
    DefectSpool.ASpool.Count;

COLLECT PercentStagedSpoolAtTime$<CurrentStepTwo>$
    100*EnteringStagedSpool/ (TotNoSpecs*4);
COLLECT NrStagedSpoolAtTime$<CurrentStepTwo>$
    StagedSpool.ASpool.Count;

IF Pulling;
COLLECT PercentFeedbackAtTime$<CurrentStepTwo>$
    100*EnteringFeedback/ (TotNoAreas*40);
COLLECT NrFeedbackAtTime$<CurrentStepTwo>$
    Feedback.ADatum.Count;
ENDIF;

COLLECT PercentWorkAreaReadyAtTime$<CurrentStepTwo>$
    100*EnteringWorkAreaReady/TotNoAreas;
COLLECT NrWorkAreaReadyAtTime$<CurrentStepTwo>$
    WorkAreaReady.AnArea.Count;

COLLECT PercentAreaDoneAtTime$<CurrentStepTwo>$
    100*EnteringAreaDone/TotNoAreas;
COLLECT NrAreaDoneAtTime$<CurrentStepTwo>$
    AreaDone.AreaThatIsDone.Count;

PRINT OutputFile "Time %f CutSheet %.4f %.0f DefectSpool %.4f %.0f
StagedSpool %.4f %.0f Feedback %.4f %.0f WorkAreaReady %.4f %.0f
AreaDone %.4f %.0f \n"
    CurrentStepTwo
    100*EnteringCutSheet/ (TotNoSpecs*4)
    CutSheet.ACutSheet.Count
    100*EnteringDefectSpool/ (TotNoSpecs*4)
    DefectSpool.ASpool.Count
    100*EnteringStagedSpool/ (TotNoSpecs*4)
    StagedSpool.ASpool.Count
    100*EnteringFeedback/ (TotNoAreas*40)
    Feedback.ADatum.Count
    100*EnteringWorkAreaReady/TotNoAreas
    WorkAreaReady.AnArea.Count
    100*EnteringAreaDone/TotNoAreas
    AreaDone.AreaThatIsDone.Count;

ASSIGN CurrentStepTwo CurrentStepTwo+TimeStepSize;

WEND;

COLLECT ProjDuration ProjectFinish;

COLLECT InstallFirstCollector Install.FirstStart;

PRINT OutputFile " + - + - + - + - + - + - + - \n";
PRINT OutputFile
"Iteration %.0f Project Finish %.4f \n"
    CurrentIteration ProjectFinish;
PRINT OutputFile " + - + - + - + - + - + - + - \n";

```

```

    ASSIGN CurrentIteration CurrentIteration+1;
    ASSIGN CurrentStepTwo 0;

WEND;

/*****
/* REPORTING
*****/
/ Now that iterations are done, print out statistics on the collectors.

PRINT StdOutput
"ProjectDuration Ave \t %8.4f \t SD \t %8.4f \t Min \t %8.4f \t Max \t %8.4f
\n"
    ProjDuration.AveVal
    ProjDuration.SDVal
    ProjDuration.MinVal
    ProjDuration.MaxVal;

ASSIGN CurrentStep 0;

PRINT StdOutput
"Item \t Time \t Ave \t SD \t Min \t Max \n";

WHILE CurrentStep<=EndTime;

    PRINT StdOutput
    "PercentCutSheet \t %6.0f \t %8.4f \t %8.4f \t %8.4f \t %8.4f \n"
        CurrentStep
        PercentCutSheetAtTime$<CurrentStep>$.AveVal
        PercentCutSheetAtTime$<CurrentStep>$.SDVal
        PercentCutSheetAtTime$<CurrentStep>$.MaxVal
        PercentCutSheetAtTime$<CurrentStep>$.MinVal;

    PRINT StdOutput
    "NrCutSheet \t %6.0f \t %8.4f \t %8.4f \t %8.4f \t %8.4f \n"
        CurrentStep
        NrCutSheetAtTime$<CurrentStep>$.AveVal
        NrCutSheetAtTime$<CurrentStep>$.SDVal
        NrCutSheetAtTime$<CurrentStep>$.MaxVal
        NrCutSheetAtTime$<CurrentStep>$.MinVal;

    PRINT StdOutput
    "PercentDefectSpool \t %6.0f \t %8.4f \t %8.4f \t %8.4f \t %8.4f \n"
        CurrentStep
        PercentDefectSpoolAtTime$<CurrentStep>$.AveVal
        PercentDefectSpoolAtTime$<CurrentStep>$.SDVal
        PercentDefectSpoolAtTime$<CurrentStep>$.MaxVal
        PercentDefectSpoolAtTime$<CurrentStep>$.MinVal;

    PRINT StdOutput
    "NrDefectSpool \t %6.0f \t %8.4f \t %8.4f \t %8.4f \t %8.4f \n"
        CurrentStep
        NrDefectSpoolAtTime$<CurrentStep>$.AveVal
        NrDefectSpoolAtTime$<CurrentStep>$.SDVal
        NrDefectSpoolAtTime$<CurrentStep>$.MaxVal
        NrDefectSpoolAtTime$<CurrentStep>$.MinVal;

```

```

PRINT StdOutput
"PercentStagedSpool \t %6.0f \t %8.4f \t %8.4f \t %8.4f \t %8.4f \n"
  CurrentStep
  PercentStagedSpoolAtTime$<CurrentStep>$.AveVal
  PercentStagedSpoolAtTime$<CurrentStep>$.SDVal
  PercentStagedSpoolAtTime$<CurrentStep>$.MaxVal
  PercentStagedSpoolAtTime$<CurrentStep>$.MinVal;

PRINT StdOutput
"NrStagedSpool \t %6.0f \t %8.4f \t %8.4f \t %8.4f \t %8.4f \n"
  CurrentStep
  NrStagedSpoolAtTime$<CurrentStep>$.AveVal
  NrStagedSpoolAtTime$<CurrentStep>$.SDVal
  NrStagedSpoolAtTime$<CurrentStep>$.MaxVal
  NrStagedSpoolAtTime$<CurrentStep>$.MinVal;

PRINT StdOutput
"PercentFeedback \t %6.0f \t %8.4f \t %8.4f \t %8.4f \t %8.4f \n"
  CurrentStep
  PercentFeedbackAtTime$<CurrentStep>$.AveVal
  PercentFeedbackAtTime$<CurrentStep>$.SDVal
  PercentFeedbackAtTime$<CurrentStep>$.MaxVal
  PercentFeedbackAtTime$<CurrentStep>$.MinVal;

PRINT StdOutput
"NrFeedback \t %6.0f \t %8.4f \t %8.4f \t %8.4f \t %8.4f \n"
  CurrentStep
  PercentFeedbackAtTime$<CurrentStep>$.AveVal
  PercentFeedbackAtTime$<CurrentStep>$.SDVal
  PercentFeedbackAtTime$<CurrentStep>$.MaxVal
  PercentFeedbackAtTime$<CurrentStep>$.MinVal;

PRINT StdOutput
"PercentWorkAreaReady \t %6.0f \t %8.4f \t %8.4f \t %8.4f \t %8.4f \n"
  CurrentStep
  PercentWorkAreaReadyAtTime$<CurrentStep>$.AveVal
  PercentWorkAreaReadyAtTime$<CurrentStep>$.SDVal
  PercentWorkAreaReadyAtTime$<CurrentStep>$.MaxVal
  PercentWorkAreaReadyAtTime$<CurrentStep>$.MinVal;

PRINT StdOutput
"NrWorkAreaReady \t %6.0f \t %8.4f \t %8.4f \t %8.4f \t %8.4f \n"
  CurrentStep
  NrWorkAreaReadyAtTime$<CurrentStep>$.AveVal
  NrWorkAreaReadyAtTime$<CurrentStep>$.SDVal
  NrWorkAreaReadyAtTime$<CurrentStep>$.MaxVal
  NrWorkAreaReadyAtTime$<CurrentStep>$.MinVal;

PRINT StdOutput
"PercentAreaDone \t %6.0f \t %8.4f \t %8.4f \t %8.4f \t %8.4f \n"
  CurrentStep
  PercentAreaDoneAtTime$<CurrentStep>$.AveVal
  PercentAreaDoneAtTime$<CurrentStep>$.SDVal
  PercentAreaDoneAtTime$<CurrentStep>$.MaxVal
  PercentAreaDoneAtTime$<CurrentStep>$.MinVal;

```



```
PRINT StdOutput
"NrAreaDone \t %6.0f \t %8.4f \t %8.4f \t %8.4f \t %8.4f \n"
  CurrentStep
  NrAreaDoneAtTime$<CurrentStep>$.AveVal
  NrAreaDoneAtTime$<CurrentStep>$.SDVal
  NrAreaDoneAtTime$<CurrentStep>$.MaxVal
  NrAreaDoneAtTime$<CurrentStep>$.MinVal;

ASSIGN CurrentStep CurrentStep+TimeStepSize;

WEND;

ENDIF;

/**** END * OF * FILE ****
```