Proceedings of the 11th North American International Conference on Industrial Engineering and Operations Management, Milwaukee, Wisconsin, USA, June 9-11, 2026

Proceedings of the International Conference on Industrial Engineering and Operations Management

Publisher: IEOM Society International, USA DOI: 10.46254/NA11.20260321
Published: June 9, 2026

An Integrated Production and Supply Chain Simulation Framework for Analysis of Complex Manufacturing Systems

Grant Robertson, Maxime Morkos and Agilan Anand
Grado Department of Industrial and Systems Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA 24060, USA
grantrobertson26@vt.edu, maximemorkos@vt.edu, agilan@vt.edu

I. Esra Büyüktahtakın
Professor
Grado Department of Industrial and Systems Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA, USA
esratoy@vt.edu

Randolph L. Bradley
Senior Technical Fellow
The Boeing Company
San Francisco, CA, USA
Randolph.L.Bradley@Boeing.com

Abstract

This project develops an integrated production–supply chain simulation framework for evaluating operational and financial trade-offs in complex manufacturing systems. In many production environments, scheduling and supply-chain planning are analyzed through disconnected models, causing task priorities to be set without timely visibility into material availability, shortage propagation, expediting needs, or inventory-policy impacts. This separation can lead to infeasible schedules, production line blockages, delayed completions, and suboptimal financial performance. To address this challenge, we couple production scheduling, inventory replenishment, material availability, and shortage-expediting decisions through a custom service-bus run-time infrastructure (RTI) that enables real-time request-response and publish-subscribe communication between simulation federates. The framework is evaluated on an anonymized 50-product manufacturing case study using combinations of due-date-based scheduling heuristics and inventory policies. Results show that integrated scheduling and inventory coordination substantially improve performance relative to the Baseline–Part-at-a-Time (PaaT) configuration. Demand Driven Material Requirements Planning (DDMRP) heuristic configurations eliminate lateness and part-shortage events, reduce traveled work by approximately 99%, and increase throughput by 184%, but require higher inventory investment. In contrast, System Availability (As) heuristic configurations achieve strong cash-flow gains and substantial lateness reductions while lowering average inventory value on hand, offering a more capital-efficient alternative. Statistical analysis confirms significant scenario effects across all reported metrics. These findings demonstrate that the proposed framework provides a practical decision-support capability for balancing production efficiency, service reliability, inventory investment, and financial performance in complex manufacturing systems.

Keywords
High-Level Architecture, Run-Time Infrastructure, Integrated Simulation, Industry 4.0, Priority Dispatching Rules

1. Introduction
This project develops an integrated production–supply chain simulation framework that couples production scheduling, inventory availability, and material expediting decisions through a service-bus RTI architecture. The framework enables stakeholders to evaluate how inventory policies affect production flow, lateness, traveled work, and financial performance. This work was motivated by the challenges observed in modern manufacturing operations, and this case study is intended to represent large-scale supply chain and production systems common to heavy industry. With limited interconnectedness between models, competing measures of success in supply chain and production simulations can lead to sub-optimal decision-making and cash flow when solutions are tailored to specific business functions. This limits a company's ability to evaluate how inventory and safety stock policies influence key financial metrics such as cash flow and ROI.

The work of this project is being built off of two previous Virginia Tech Industrial and Systems Engineering (ISE) senior design projects: Boeing Supply Chain Digital Twin Simulation and Optimizing Task Prioritization in Boeing 787 Final Assembly. The results of the Boeing Supply Chain Digital Twin Simulation assess the advantages and tradeoffs of Push, Pull, Constant Work-in-Progress, and Demand Driven Material Requirements Planning for inventory management (Ganguly et al. 2025). Optimizing Task Prioritization in Boeing 787 Final Assembly was aimed at improving the sequencing of tasks to decrease production lateness (Pannell et. al 2025). Instead of considering a specific commercial aircraft program, this project is based on an anonymized production scenario for the HokieBird air taxi, named in honor of Virginia Tech's school mascot. The anonymized and simplified supply chain and production data for the HokieBird reflect generic production lines found across heavy industry globally

1.1 Objectives
This project aims to provide a framework for integrating simulation models to assess the impact of supply chain inventory policies on production systems. Additionally, this work will evaluate the performance of four single-machine scheduling heuristics in a multi-stage production system with unidirectional production flow and precedence-based constraints.

2. Literature Review
In this section, we review literature on simulation model integration standards, methods for data transfer across parallel and distributed systems, and research on the single-machine total tardiness scheduling problem.

High level architecture (HLA) defines a common set of rules for federates governing the behaviors and restraints of simulation models (Li et al. 2021). The HLA acts as a structure which allows for simulation models to be used for purposes other than their original functionality. The HLA premise is built on the idea that one single simulation model cannot fulfill all possible needs, and thus, models should be implemented with the goal of reuse (Dahmann 1997). In addition to defining the set of rules for simulation federates, the HLA is also comprised of an object model template (OMT) which defines how data will be formatted for use between federates, and a runtime infrastructure (RTI) which facilitates communication between federates as a centralized data hub (Fujimoto 1998).

IEEE (2010) defines the five rules for the federation and each federate, which ensure that federates are modular in function. The federation must have a federate object model (FOM) documented in accordance with the HLA OMT. Object instance attributes must be maintained by the federates, not the RTI, all data must be exchanged via the RTI instead of directly between the federates. Additionally, federates must interact with the RTI as per the rules set by the HLA, and each object attribute may be owned by at most one federate. Similarly, five rules define the behavior of individual federates. Each federate must have a simulation object model (SOM) documented in accordance with the HLA OMT, and must be able to update and send/recieve updates to their instances as defined in their SOM. Federates may transfer or accept ownership of instance attributes based on these definitions. The conditions under which state updates are provided are governed by the FOM, while each federate maintains control over its own local time in a way that enables coordinated data exchange with other federates.

Tolk and Muguira (2003) defines six levels of interoperability between systems in the Levels of Conceptual Interoperability Model (LCIM), labeled from Level 0 to Level 5. Level 0 represents no interoperability, where there is no connection between systems. Level 1 corresponds to technical interoperability, where a basic means to exchange data between systems exists. Level 2, syntactic interoperability, ensures that data is structured the same way across systems. Level 3, semantic interoperability, allows systems to share a common understanding of the meaning of exchanged data. Level 4, pragmatic interoperability, defines how that data is used within each system. Finally, Level 5, conceptual interoperability, enables systems to understand the state changes that occur in other systems. Technical interoperability (LCIM 1) can be achieved via coupling models; this can take the form of pairwise coupling directly between models or via service-bus coupling. Service-bus coupling takes a centralized form where data is exchanged from models to a data broker that distributes data to the appropriate models (Huiskamp and van der Berg 2017).

Mehl and Hammes (1993) discussed two methods for sharing information across a distributed simulation. The first are read requests; the requestor sends a request to the owner of the information and pauses the requestor's operations until after it receives the data. The second are update requests; the owner of new information sends an updated value to the owner of the outdated information without stopping its own operations. Fujimoto (1996) categorizes messages across a parallel simulation: timestamp orders and receive orders. Gan et al. (2003) developed two new methods for transferring information through a parallel distributed simulation while relaxing the lookahead constraint and reducing execution time of the simulation. Those methods were PushRO and PullRO; the timestamp sequenced orders of the previous methods are replaced by request order (RO) interaction updates that may be handed off in any order. Low et al. (2006) presented two improvements over PushRO and PullRO that offer a time guarantee over their RO counterparts: PushROTG and PullROTG. The experiment utilized varying request ratios between the owner and reader models and different lookahead values; PullRO and PullROTG showed reductions in execution time over PushRO and PushROTG when request ratios and lookahead values are high.

Bradley and Goentzel (2012) demonstrate the impact of integrating a supply chain simulation with an inventory optimization model to evaluate stocking policies for different system level fill rate goals. A common industry approach of optimizing fill rates for two separate groups split by part price, using a greedy heuristic to optimize inexpensive parts to a higher fill rate and expensive parts to a lower fill rate, showed many expensive parts went underfilled, leaving inventory unbalanced (Sherbrooke, 2004). Bradley and Goentzel (2012) then present five distinct segments based on part consumption and price to balance the distribution of inventory purchases.

Although the problem space being observed is a multi-stage scheduling problem with precedence constraints, sequencing decisions at each build position can be represented as independent single-machine scheduling problems. Li (2001) breaks down the multiple machine job scheduling problem into multiple single-machine scheduling problems running in parallel. In the single-machine total tardiness (SMTT) problem, a set of n jobs are available to schedule on a single machine which can work on at most one job j with processing time p_j and due date d_j at a given time; total tardiness is minimized under the objective function $\min \sum T_j$, where $T_j = \max(0, C_j - d_j)$, and C_j is the completion time of job j (Koulamas 2010). Bigras et al. (2008) demonstrated the similarities between the single-machine scheduling problem and the time-dependent traveling salesman problem.

Many heuristics have been proposed for solving the single-machine problem. Jackson (1955) presented the earliest due date (EDD) task prioritization rule which minimizes the maximum lateness of all jobs; the lateness of a job j is defined as $C_j - d_j$. Baker and Bertrand (1982) introduced the modified due date (MDD) task prioritization rule which appends tasks to the set of scheduled tasks, S, in ascending order of each task's modified due date, $d_j^*(t) = \max(t + p_j, d_j)$. Recently, Çetinkaya et al. (2025) proposed two challenger LLM derived heuristics, MDD Challenger (MDDC) and EDD Challenger (EDDC), which outperform their known heuristics. MDDC and EDDC were algorithmically created by LLMs over 10,000 iterations intended to reduce the optimality gap over their known counterparts. The MDDC introduces a scaled urgency factor and system congestion term when calculating $d_j^*(t)$. The EDDC performs a conditional swap of the last two scheduled tasks on a given day if swapping them would prevent downstream delays.

The reviewed literature discusses system interoperability, single-machine task sequencing heuristics, and inventory optimization; however, research is limited on integrating these three fields into one system architecture capable of determining the impact of material availability and scheduling decisions on financial performance. This gap motivates the integrated production and supply chain simulation framework created in this project.

2.1 Key Contributions

Complex manufacturing systems require coordinated decisions across production scheduling, material availability, inventory replenishment, and shortage response. Yet these functions are often modeled separately, limiting the ability to evaluate how inventory policies affect production flow, delay propagation, and financial performance. This paper addresses this gap by developing an integrated production–supply chain simulation framework connected through a custom service-bus Run-Time Infrastructure.

The contributions are threefold. First, we develop a modular federated simulation architecture that enables real-time coordination between production scheduling, supply-chain inventory, and expediting components through request-response and publish-subscribe communication. Second, we embed material availability directly into task sequencing, allowing scheduling decisions to account for on-hand inventory, delayed materials, and shortage-driven disruptions. Third, we demonstrate the value of the framework through a 50-product manufacturing case study that quantifies operational and financial trade-offs across scheduling and inventory policy combinations. DDMRP-based heuristic configurations eliminate average lateness and shortage events, reduce traveled work by approximately 99%, increase annualized throughput by 184% relative to the Baseline–PaaT configuration, and increase in average inventory value on hand by 5.1%. In contrast, As-based heuristic configurations provide a more capital-efficient alternative, reducing average lateness by 95%, increasing annualized cash flow by approximately 68–69%, and reducing the average value of on hand inventory by more than 45%.

Together, these contributions provide a practical decision-support framework for evaluating trade-offs between production efficiency, service reliability, inventory investment, and financial performance in complex manufacturing systems.

3. Methods

This project utilizes an integrated supply chain and production simulation to assess the impact of inventory policies on key production performance indicators, such as average lateness, the average number of traveled tasks (tasks that miss their due date and are pushed to a subsequent build position), and cash flow. A federated simulation architecture was developed to couple supply chain and production simulations to an HLA-compliant service-bus RTI. The framework enables interoperability between simulation models, allowing production demands and material availability to be seamlessly shared. Experimental scenarios were created using varying production models and inventory policies.

3.1 Integrated Production and Supply Chain Simulation Architecture

The simulation is integrated as a federated architecture in which production scheduling, supply-chain ordering, shortage expediting, and demand are separated into modular, interacting components. A Python orchestration layer initializes the federation, instantiates the production scheduler and supply chain VBA-backed adapter, and translates run outputs to an SQLite database. Communication between models is handled via a custom HLA inspired service-bus RTI, allowing publish-subscribe interactions and synchronous request-response exchanges between federates. Figure 1 represents the system architecture of how data moves through the integrated system with respect to the HLA/RTI service-bus. In the integrated framework, the production model attempts to schedule tasks and requests parts from on hand inventory from the Supply Chain Digital Twin, indicated as the supply chain model in the figure. The supply chain federate then returns the feasible and delayed tasks to the production model, informing which of the requested tasks have the materials and thus can be scheduled. The expedite optimizer reacts to confirmed part shortages as they occur using the impact to the critical path length as the delay. Passive observers record state snapshots of simulation events to an SQLite database for post-simulation analysis. The SQLite tables are then sent for data logging and analytics to be used in post-simulation visualization. The integration layer synchronizes production demand, material availability, shortages, and completion events across the production scheduler and BLISS supply-chain model. SQLite is used for persistent logging and post-simulation analysis, while the HLA/RTI service-bus enables real-time request-response and publish-subscribe communication during simulation execution.

Table 1 contains each of the object classes represented in the integrated simulation, along with the federate that owns it. Additionally, each of the federates own unique interaction classes. For instance, Production controls interaction classes such as CheckPartAvailability, ConsumePartsBatch, ReleaseUnconsumedReservations, TaskStatusUpdates, TasksCompleted, and ProductCompleted which all are distributed to other simulation models that subscribe to the particular interaction class. SupplyChain controls the interaction classes for SupplyStatusUpdated and PartShortagesDetected that are relayed to the Production as they are updated.

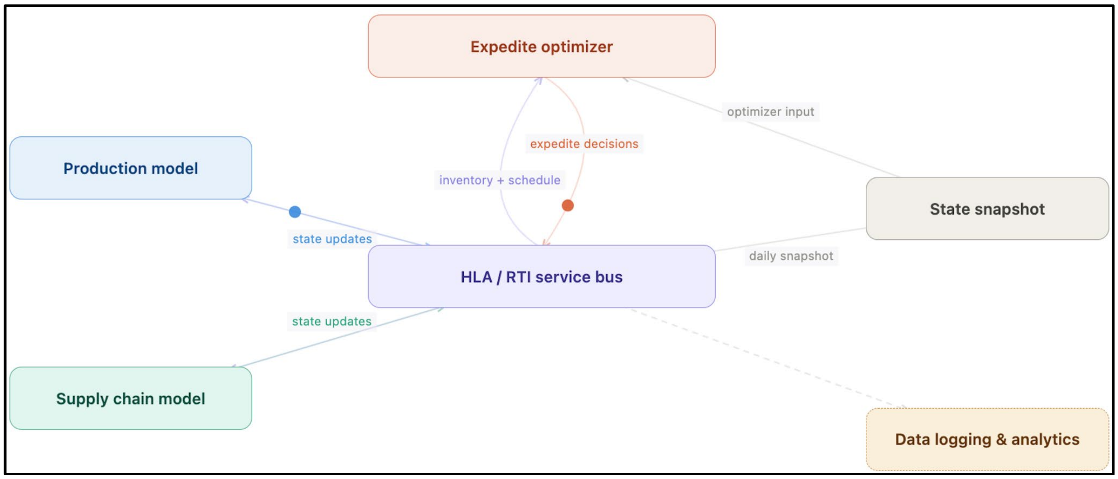

Figure 1. System architecture diagram of data linkages between simulation models and the service-bus RTI

Table 1. Object class ownership and description of classes

Object Class	Owning Federate/Model	What it Represents
Production Task	ProductionSchedulerFederate	Task-level production execution state
Inventory Item	SupplyChainFederate	Part-level inventory and replenishment state
Shortage Record	SupplyChainFederate	Active part shortages blocking tasks
Expedite Decision	ExpediteOptimizerFederate	Expedite/emergency-order decisions

3.2 Inventory Policies
The Supply Chain Digital Twin Simulation in the federation handles reordering decisions based on current on-hand inventory, reorder points, reorder quantities, and dynamic buffers. A range of standard inventory policies can be simulated; the three policies compared in this study were Part-at-a-Time Optimization, Global System Availability Inventory Optimization, and Demand Driven Material Requirements Planning.

3.2.1 Part-at-a-Time (PaaT) Optimization
The baseline inventory policy determines the Reorder Point / Reorder Quantity (ROP/ROQ) by conducting a part-at-a-time inventory optimization to minimize total ordering, holding, and backorder costs when demand follows a gamma distribution (Tyworth 2000). The resulting ROP/ROQ policies result in fill rates which minimize total costs for each part at a single stocking location.

3.2.2 Global System Availability (As) Inventory Optimization
The System Availability (As) model determines a reorder point/reorder quantity (ROP/ROQ) policy for each part through a global inventory optimization conducted at a single stocking location. Its purpose is to minimize total inventory investment while satisfying a prescribed system availability target, given ordering, holding, and shortage costs and assuming that demand follows a gamma distribution. Using the PaaT solution as the initial condition, the As model evaluates each part sequentially to estimate the system-level change in expected backorders associated with a one-unit increase in its reorder point, which is equivalent to increasing safety stock. Applying marginal analysis, the model then selects the part that yields the greatest reduction in expected backorders per incremental unit of inventory investment. This iterative process continues until the specified system availability target is attained (Sherbrooke 2004).

3.2.3 Demand Driven Material Requirements Planning (DDMRP) Part-at-a-Time Heuristics
The DDMRP model uses the heuristics defining this system's three-zone color-coding system (Red, Yellow, and Green) to set inventory levels based on the actual demand rather than forecasts, without attempting to set strategic decoupling points in production to reduce the bullwhip effect, and without separating spike demand for spares from production demand (Ptak 2019).

3.3 Single-Machine Scheduling Heuristics
The production simulation decomposes the larger multi-team four build-position flow-line simulation into several single-machine scheduling problems running in parallel. Dispatching rules are applied to each of the 17 production teams, utilizing their time resource as the constraint for the number of tasks that may be scheduled. Each simulation day, eligible tasks are filtered by precedence and part availability; when multiple tasks compete for the time resources of a team, single-machine scheduling rules rank the set of unscheduled tasks to assign order.

Description of parameters:

C_i : Completion time of job i
d_j : Due date of task j
p_j : Processing time of task j
t : Current time
T_{comp} : Completion time
μ : Modified due date score
θ : Scaled adjustment due date weight
σ : System congestion factor
ρ : Processing time urgency factor
S : Set of scheduled tasks
U : Set of unscheduled tasks
J : Set of all tasks

3.3.1 Earliest Due Date
The earliest due date scheduling heuristic minimizes maximum lateness by scheduling tasks with the nearest due date first (Jackson 1955).

Algorithm 1 Earliest Due Date (EDD)

Sort jobs in nondecreasing order of due date:

$$S \leftarrow \mathrm{sort}(J \text{ by } d_j)$$

$t \leftarrow 0$
for $i = 1$ **to** $|S|$ **do**
  Process job $S[i]$ at time t
  $t \leftarrow t + p_{S[i]}$
end for

3.3.2 Earliest Due Date Challenger (EDDC)
The EDDC algorithm refines the EDD schedule by determining if a conditional swap of the last two tasks in S would prevent any future delays if the due date of the second to last task is after the last task and if the second to last task's d_j is after the T_{comp} of the currently scheduled jobs (Çetinkaya et al. 2025).

Algorithm 2 EDD Challenger (EDDC)

Sort jobs by d_j, then by p_j for jobs with equal due dates
for $i = 1$ **to** $|S|$ **do**
  $j \leftarrow i$
  while $j > 1$ **and** $d_{S[j]} < d_{S[j-1]}$ **do**
    $T_{\mathrm{comp}} \leftarrow \sum p_{S[1:j]}$ ▷ Completion time of jobs up to position j
    if $j = |S|$ **and** $d_{S[j-1]} > T_{\mathrm{comp}}$ **then**
      Swap $S[j]$ and $S[j-1]$ to avoid delay
    end if
    $j \leftarrow j - 1$
  end while
end for

3.3.3 Modified Due Date (MDD)
The MDD algorithm considers the task processing time as well as the task due date when scheduling to prioritize tasks that are at risk of being late (Baker and Bertrand 1982).

Algorithm 3 Modified Due Date (MDD)

$t \leftarrow 0$
while $U \neq \emptyset$ **do**
  Select job j^* with the smallest $\max(d_j,\ t + p_j)$ from U
  $t \leftarrow t + p_{j^*}$, remove j^* from U
end while

3.3.4 Modified Due Date Challenger (MDDC)

The MDDC algorithm refines MDD by adding a scaled urgency factor and system congestion term for its scheduling decisions. The processing time urgency factor is based on the task's relative processing time when compared to the maximum and mean processing times of other tasks in *U* (Çetinkaya et al. 2025).

Algorithm 4 MDD Challenger (MDDC)

$t \leftarrow 0$, initialize S as empty, $U \leftarrow$ all jobs
Sort U by p_j in ascending order
Initialize p_U and d_U as the processing times and due dates of all unscheduled jobs
while $U \neq \emptyset$ **do**
  $\mu \leftarrow \max(p_U \times 1.1 + t,\ d_U)$ ▷ Base modified due date score
  $\rho \leftarrow \min\left(\frac{p_U}{t + \max(p_U)},\ 1\right)$ ▷ Processing time urgency factor
  $\theta \leftarrow \frac{\rho^2}{1 + \rho^2}$ ▷ Scaled adjustment weight
  $\mu \leftarrow \mu \times (1 + \theta)$ ▷ Apply urgency adjustment to score
  $\sigma \leftarrow \frac{p_U}{t + \sum(p_U)/|U|}$ ▷ Workload pressure factor
  $\mu \leftarrow \mu + \sigma$ ▷ Apply workload adjustment to score
  Select job with minimum μ score and append to S
  Update U, t, p_U, and d_U by removing scheduled job
end while

3.4 Experimental Design

Baseline performance was determined using the Baseline production model and the PaaT inventory policy. The Baseline production model simulates current day behavior of operators on the production floor; each task's processing time is multiplied by an inefficiency factor of [1.0, 3.1] and a daily chaos factor in [0.8, 1.6], and uses a random priority score that prioritizes shorter tasks that are in the correct build position.

Each scheduling policy was tested against the PaaT, As, and DDMRP inventory policies to determine how inventory policy affects financial metrics such as cash flow when simulated in a real production scenario, and how scheduling policies compare when a multi-stage machine problem is decomposed into many parallel single-machine scheduling problems with limited decisions each day.

The run file instantiates a 50-product production horizon and runs until all products are delivered. All numerical results were normalized against the Baseline-PaaT policy combination. Throughput, cash flow, and part shortages were annualized to represent the respective data over one year, due to the differing time horizons of each scenario. All scenarios were averaged over ten replications with lead times and task standard time being in [0.75, 1.25] and [0.80, 1.60] respectively.

A two-sided Welch t-test on replication-level results was conducted for ten replications per scenario to determine if there was a significant statistical difference ($p < 0.05$) between the metric-level results and the Baseline-PaaT scenario results.

4. Data Collection

Table 2 presents a mapping of the simulation's data by categorizing input parameters into distinct operational, financial, and logical domains. By synchronizing foundational factory constraints from the Source Data workbook with runtime variables, the layout identifies specific model variables, their default baseline values, and their exact sources. This structured format distinguishes between static workbook data and parameters injected at runtime. Consequently, the table validates the origin of all factory constraints and inventory policies, showing a transparent review of the assumptions used to evaluate performance metrics.

Table 2. Model Input Parameters and Data Sources

Category	Parameter	Default Value / Range	Data Source / Reference
Production: Operational	Task Dependencies	Varied per aircraft	task_relationships sheet
	Task Standard Time	Varied per task	task_status_per_product sheet
	Delivery Windows	Varied per product	schedule sheet
	Team Staffing	Mechanics per team	team_work_schedule sheet
Production: Capacity	Utilization Cap	85%	Heuristics Schedulers
	Baseline Team Limits	4.0 to 20.0 hrs/day	Baseline_ProductionScenario.py
Production: Inefficiency	Traveled Work Multiplier	+0.60 (Stacks)	Baseline_ProductionScenario.py
	Team 3A/3B/3C Penalty	+0.40 (Stacks)	Baseline_ProductionScenario.py
	Daily Chaos Factor	0.8 to 1.6 (Random)	Baseline Multipliers
Production: Financial	Labor Rate	$58.75 / hour	Run_File.py (CLI Argument)
	Delay Penalty	$10,000 / day	Run_File.py (Runtime Default)
	Expedite Cost	$150 / unit	Expedite Optimizer Hardcode
Supply Chain	Master Parts & Inventory	Comprehensive list	BLISS supply chain workbook
	Policy Tables	PaaT, DDMRP	Policy_Selection_Options
	External Refinements	Optimization Data	Inventory Optimization & Source Data Imports
	Zero Demand Filter	Automatic removal	Dataset Ingestion Clean-up
Integration Layer	Demand Signal Path	ProductionDemand	Named Ranges (Python-Excel)
	Availability Check	Synchronous Request	Simulation Service-Bus
	Consumption / Completion	Asynchronous Publishes	Simulation Service-Bus
	Post-Run Analytics	SQLite Tables	integration_log / Side Channels

5. Results and Discussion

5.1 Numerical Results

As shown in Table 3, the EDDC-DDMRP configuration achieved the strongest operational reliability, reducing average lateness by 100% and traveled work by 99.6% relative to the Baseline-PaaT configuration. It also eliminated part-shortage events and increased annualized cash flow by 62.9%. However, these gains came with a 5.1% increase in average inventory value on hand, indicating that DDMRP improves reliability partly through higher inventory investment. In contrast, the EDDC-As configuration provides a more capital-efficient alternative. It reduced average lateness by 95.0%, reduced traveled work by 37.9%, increased annualized cash flow by 68.9%, and reduced average inventory value on hand by 45.3%. Although it does not eliminate shortages as DDMRP does, it substantially improves performance while using inventory more efficiently. The percentage improvements should be interpreted relative to the intentionally degraded baseline scenario with stochastic inefficiency factors observed in a real-world manufacturing environment. As a result, large percentage gains and reductions reflect both the effectiveness of the production and inventory policy decisions as well as the extent to which the baseline scenario is constrained by operational inefficiency. Therefore, the values measure improvement relative to the modeled current-state production environment and not an optimized manufacturing system.

These results demonstrate the value of the integrated simulation framework: it enables decision-makers to evaluate not only production performance, but also the trade-offs between lateness, traveled work, cash flow, shortage events, and inventory investment. The results also show that task sequencing plays an important role in production performance. For example, even when paired with the PaaT inventory policy, EDDC reduces lateness by 87.8%, while Baseline-As reduces lateness by 60.9% relative to Baseline-PaaT.

Additionally, the results of Table 3 suggest a practical decision rule for inventory policy selection. Manufacturers should utilize DDMRP when late deliveries carry high monetary or customer-service penalties, as DDMRP shows very strong operational performance in reducing lateness and traveled work by increasing the available on-hand material quantities. However, manufactures should utilize the As policy when moderate delays are acceptable and warehouse capacity is constrained, because As provides stronger operational performance than the common PaaT policy without relying on overstocking materials.

An interesting finding within Table 3 is that all DDMRP combinations show nearly identical average lateness, average traveled work, and annualized throughput. This suggests that when the DDMRP ROP/ROQ buffers provide consistently high material availability, the main limiting constraint of production shifts from material availability to manufacturing-team capacity. As the dispatching rules are applied on a production-team level, the small number of feasible tasks available to each team leads the dispatching rules to generate nearly identical schedules when material starvation is negligible. The convergence of results under the DDMRP policy does not indicate that the production dispatching rules are equivalent, but rather, it reflects the limited number of scheduling choices available to each production team when material availability is high and jobs are not delayed.

A ±20% one way sensitivity analysis of labor costs, delay penalties, and expedite costs per unit over ten replications revealed that final cashflow was most sensitive to changes in labor costs (±15.3%), less sensitive to fluctuations in delay penalties (±4.5%), and differences to expedite costs were negligible (±0.01%) when compared to the nominal EDDC-As scenario over ten replications.

Table 3. Performance metrics for production scheduling and inventory policy combinations in a 50-product scenario, normalized to the Baseline–PaaT configuration and averaged over ten replications.

Model	Average Lateness	Average Traveled Work	Annualized Throughput	Annualized Cash Flow	Annualized Part Shortage Events	Average Inventory Value on Hand
Baseline-PaaT	0%	0%	0%	0%	0%	0%
Baseline-As	-60.9%*	+1.5%	+56.6%*	+19.3%	-54.9%*	-21.1%*
Baseline-DDMRP	-59.6%*	+1.2%	+58.4%*	-11.4%	-100.0%*	+33.5%*
EDD-PaaT	-88.9%*	-21.2%*	+113.3%*	+65.0%*	+52.5%*	-43.1%*
EDD-As	-94.9%*	-37.9%*	+141.4%*	+68.8%*	-5.7%	-45.1%*
EDD-DDMRP	-100.0%*	-99.6%*	+184.0%*	+62.4%*	-100.0%*	+6.3%
EDDC-PaaT	-87.8%*	-21.2%*	+111.3%*	+63.4%*	+52.4%*	-42.2%*
EDDC-As	-95.0%*	-37.9%*	+143.3%*	+68.9%*	-10.5%*	-45.3%*
EDDC-DDMRP	-100.0%*	-99.6%*	+184.0%*	+62.9%*	-100.0%*	+5.1%

Model	Average Lateness	Average Traveled Work	Annualized Throughput	Annualized Cash Flow	Annualized Part Shortage Events	Average Inventory Value on Hand
MDD-PaaT	-88.8%*	-21.2%*	+113.3%*	+64.9%*	+54.1%*	-43.1%*
MDD-As	-93.0%*	-38.1%*	+142.8%*	+69.2%*	-3.3%	-45.3%*
MDD-DDMRP	-100.0%*	-99.6%*	+184.0%*	+62.8%*	-100.0%*	+5.7%
MDDC-PaaT	-87.8%*	-21.1%*	+111.3%*	+63.7%*	+52.5%*	-43.5%*
MDDC-As	-95.0%*	-40.3%*	+141.7%*	+68.9%*	-5.9%	-45.1%*
MDDC-DDMRP	-100.0%*	-99.4%*	+184.0%*	+62.6%*	-100.0%*	+6.1%
Overall scenario effect p-value	$p < 0.001$	$p < 0.001$	$p < 0.001$	$p < 0.001$	$p < 0.001$	$p < 0.001$

Note: (*) denotes significant statistical difference from Baseline-PaaT in the metric ($p < 0.05$) using a two-sided Welch t-test on replication-level results for ten replications per scenario. The bottom row reports Welch one-way ANOVA p-tests for the overall scenario by metric.

5.2 Graphical Results

Figure 2 compares the cash flow revenues of the EDDC and Baseline integrated models normalized against the Baseline PaaT in blue. In the figure all three EDDC integrated models perform similarly in the 50-product scenario; however, when extrapolated over a set extended time period, such as one year, DDMRP tends to pull ahead due to its ability to complete an increased number of products over other inventory policies due to its high on-hand inventory.

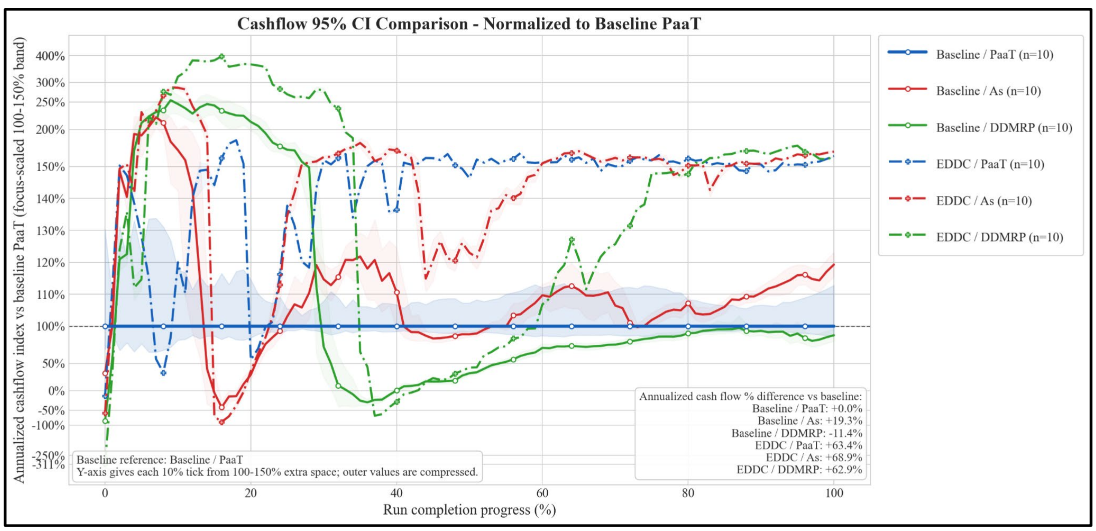

Figure 2. Cash flow diagram of EDDC and Baseline integrated models normalized against the Baseline-PaaT configuration for a 50-product production period

6. Conclusion

This paper develops and validates an integrated production and supply chain simulation framework using a service-bus RTI infrastructure. Results show improvements in key performance indicators in several scenarios, while also highlighting trade-offs across different system configurations. For the scenarios tested, the EDDC-As model shows the strongest average performance. In this model, average lateness, traveled work, and annualized part shortage events decreased by 95.0%, 37.9%, and 10.5% respectively, as shown in Table 3. Similarly, annualized throughput and cash flow increased by 143.3% and 68.9% respectively compared to the baseline model. However, several other scenarios did not perform as well and required higher inventory investments or resulted in increased shortage events, showing how improvements are not uniform across all cases.

The main takeaway is that the framework allows direct comparison of trade-offs between scheduling performance, throughput, cash flow, shortages, and inventory investments across different scheduling and inventory policy combinations. Moreover, the key contribution to this work is the integration of production scheduling, inventory management, and expediting decisions within a single real time simulation environment. Future work should extend the framework to additional production settings, increase the number of simulation runs, as well as incorporate more detailed uncertainty and cost modeling.

Acknowledgements

We thank the previous Virginia Tech teams whose work on the production and supply chain models helped establish the baseline for this project. We also thank the Penn State teams for their work on supply chain disruptions to critical materials, which supported the development of this study. Finally, we would like to thank Dr. Natalie Cherbaka for her assistance with editing and revising this paper as well as her feedback on presentation materials.

References

Baker, K. R. and Bertrand, J. W. M., A dynamic priority rule for scheduling against due-dates, *Journal of Operations Management*, vol. 3, no. 1, pp. 37-42, 1982.

Bigras, L. P., Gamache, M. and Savard, G., The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup times, *Discrete Optimization*, vol. 5, no. 4, pp. 685-699, 2008.

Bradley, J. and Goentzel, J., Simulating tomorrow's supply chain today, *Proceedings of the 2012 Winter Simulation Conference*, pp. 1-12, Berlin, Germany, December 9-12, 2012.

Çetinkaya, İ. O., Büyüktahtakın, İ. E., Shojaee, P., & Reddy, C. K. (2025). Discovering heuristics with Large Language Models (LLMs) for mixed-integer programs: Single-machine scheduling. *Computers & Operations Research, 186*, 107325. doi:10.1016/j.cor.2025.107325

Dahmann, J. S., High level architecture for simulation, *Proceedings of the First International Workshop on Distributed Interactive Simulation and Real Time Applications*, pp. 9-14, Orlando, FL, USA, January 9-10, 1997.

Fujimoto, R. M., Time management in the high level architecture, *SIMULATION*, vol. 71, no. 6, pp. 388-400, 1998.

Gan, B. P., Low, K. L., Wei, J., Wang, X., Turner, S. J. and Cai, W., Synchronization and management of shared state in HLA-based distributed simulation, *Proceedings of the 2003 Winter Simulation Conference*, vol. 1, pp. 847-854, New Orleans, LA, USA, December 7-10, 2003.

Ganguly, I., Ingham, R., Jeong, S. and Yoo, Y., Supply chain digital twin analysis, Virginia Polytechnic Institute and State University, Blacksburg, USA, 2025.

Huiskamp, W. and van der Berg, T., Federated simulations, *Studies in Systems, Decision and Control*, vol. 90, no. 6, pp. 109-137, 2017.

IEEE Computer Society, IEEE standard for modeling and simulation (M&S) high level architecture (HLA)—framework and rules, *IEEE Std 1516-2010*, August 2010.

Jackson, Scheduling a Production Line to Minimize Maximum Tardiness, Ph.D. Thesis, University of California, Los Angeles, USA, 1955.

Koulamas, C., The single-machine total tardiness scheduling problem: Review and extensions, *European Journal of Operational Research*, vol. 202, no. 1, pp. 1-7, 2010.

Li, H., Lu, J., Zheng, X., Wang, G. and Kiritsis, D., Supporting digital twin integration using semantic modeling and high-level architecture, *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems*, pp. 228-236, Springer, 2021.

Li, J., A parallel approach for solving a multiple machine job sequencing problem, Master's Thesis, University of Regina, Regina, Saskatchewan, 2001.

Low, M. Y. H., Gan, B. P., Wei, J., Wang, X., Turner, S. J. and Cai, W., Shared state synchronization for HLA-based distributed simulation, *SIMULATION*, vol. 82, no. 8, pp. 511-521, 2006.

Mehl, H. and Hammes, S., Shared variables in distributed simulation, *Proceedings of the Principles of Advanced and Distributed Simulation*, pp. 68-75, San Diego, USA, May 16-19, 1993.

Pannell, M., Chen, A., Ho, J. and Litman, A., Optimizing task prioritization in Boeing 787 final assembly, Virginia Tech, Blacksburg, USA, 2025.

Ptak, C. and Smith, C., *Demand Driven Material Requirements Planning (DDMRP)*, 3rd Edition, Industrial Press, 2019.

Sherbrooke, C. C., *Optimal Inventory Modeling of Systems*, 2nd Edition, Springer, New York, 2004.

Tolk, A. and Muguira, J. A., The levels of conceptual interoperability model, *Proceedings of the Fall Simulation Interoperability Workshop*, Orlando, USA, September 2003.

Tyworth, J. E. and Ganeshan, R., A note on solutions to the (Q, r) inventory model for gamma lead-time demand, *International Journal of Physical Distribution & Logistics Management*, vol. 30, no. 6, pp. 534-539, 2000.

Biographies

Grant Robertson is an undergraduate student of the Grado Department of Industrial and Systems Engineering at Virginia Polytechnic Institute and State University. He will be starting full-time work at The Boeing Company as a systems engineer following his May 2026 graduation. Grant has previously presented research on using LLMs to model cyber-resilient offshore wind farms at the 2025 Hume Center for National Security and Technology Colloquium. His research interests are in digital engineering and digital twins, and their applications to complex system modeling and decision support.

Maxime Morkos is an undergraduate student in the Grado Department of Industrial and Systems Engineering at Virginia Tech, with a minor in Data and Decisions. He will be starting full-time at General Motors in the TRACK rotational program following his May 2026 graduation. Maxime has experience in data-driven system analysis through research and applied projects, including work as an undergraduate research assistant analyzing data for the safety implementation of augmented reality head-up display systems in vehicles. He has also served as an undergraduate teaching assistant for introductory calculus. In addition, he has experience in data management and analytics through coursework and project based work, focusing on system modeling and performance evaluation. His interests include digital engineering, simulation, and the application of data analytics to complex systems.

Agilan Anand is an Industrial and Systems Engineering student within the Grado Department at Virginia Polytechnic Institute and State University. Upon his graduation in May 2026, he will join the American Honda Motor Company as a Manufacturing Engineer. Agilan has diverse industry experiences, including manufacturing engineering internship roles at Altec and KIND, an operations internship with Bonduelle Fresh Americas, and an engineering internship with Tekion Corp. Agilan also served as the Mechanical Lead for SolarCar at Virginia Tech. His technical expertise is further bolstered by several professional credentials, including a Microsoft Project Management Specialization, a Manufacturing Service Operations Professional Certificate (CSCMP), and SOLIDWORKS Associate and Expert certifications.

Dr. I. Esra Büyüktahtakın is a Full Professor of Operations Research in the Grado Department of Industrial and Systems Engineering at Virginia Tech, where she directs the Systems Optimization and Machine Learning Lab (SysOptiMaL). Her research focuses on multi-stage stochastic mixed-integer programming (M-SMIP) and pushes the boundaries of integrated machine learning (ML), optimization, and artificial intelligence (AI) for high-stakes decision-making under uncertainty. She develops scalable, intelligent algorithms by combining optimization theory, deep learning, and advanced computational tools to address large-scale combinatorial problems. At SysOptiMaL, Dr. Büyüktahtakın mentors students engaged in cutting-edge research on AI-driven multi-stage optimization, uniting theoretical rigor with innovative algorithm design and computational analysis. She is the recipient of the 2016 NSF CAREER Award. She has graduated 8 Ph.D. students now in leading academic and industry roles and currently advises eight more.

Randolph Bradley received his Master of Engineering in Logistics and Supply Chain Management at the Massachusetts Institute of Technology, U.S.A., in 2012. As a Senior Technical Fellow within the Supply Chain Management organization of The Boeing Company's Defense, Space & Security business, his primary role is to enable the transition from successful proposal strategy to flawless program execution, focusing on delivering value across a spectrum of supply chain business opportunities. He is experienced with requirements definition to provide focused solutions, statistical demand forecasting to estimate future requirements, strategic inventory optimization to minimize investment, simulation to reduce risk and predict performance over time, tactical asset management to enable efficient daily operations, and inventory management to warehouse assets.