# Predicting Business Process Performance with 'Real World' Queuing

**Joseph S. DeKeyrel**
**Raytheon Technical Services Company**
**Orlando, FL**
**Joseph_S_DeKeyrel@Raytheon.com**

**Stephanie J. Lackey**
**Institute for Simulation and Training**
**Orlando, FL**
**SLackey@ist.ucf.edu**

## ABSTRACT

Given a workflow system used to process orders for services, the authors assert that a faithful model of the system, loaded to reflect the actual system's state, can then be used to predict performance. Building faithful models of processes with high degrees of uncertainty can be very challenging, especially where this uncertainty exists in terms of processing times, queuing behavior and rework. Most of the literature focuses on predicting system-level performance where the servers in the system exhibit standard queuing. The authors will instead present the theory and methodology for predicting performance for an individual job in an environment where the queuing behavior is not standard. The context that the authors will use to address the aforementioned uncertainty is a multi-tiered workflow system used to accept orders for training services and return proposals against those orders. The authors will specifically explore the use of machine learning and embedded discrete event simulations to analyze and predict individual job due dates.

## ABOUT THE AUTHORS

**Dr. Joseph DeKeyrel** is the Chief Engineer for the Warfighter Support Services business unit of Raytheon Technical Services Company. After serving as a US Army Combat Engineering officer, Dr. DeKeyrel transitioned to a career as a defense contractor spending over seven years at the National Training Center analyzing, designing, and implementing hardware and software solutions for the Instrumentation System. Subsequently, he was transferred to the Orlando area where he has served as the senior technical representative on the Live Training Support Services and Warfighter FOCUS programs. Since transitioning to Central Florida, he has been engaged in research on the use of embedded modeling to predict the performance of business processes which are supported by automated workflow management systems. He earned his Ph.D from the Industrial Engineering and Management Systems department of the University of Central Florida.

**Dr. Stephanie Lackey** is the Director of the Applied Cognition & Training in Immersive Virtual Environments Lab at the University of Central Florida's Institute for Simulation and Training where she researches methods to improve simulation-based training technologies and robotic systems through the application of established and emerging trends in systems engineering and human systems integration. Dr. Lackey joined the ACTIVE Lab in 2008 following seven years of Government service with the U.S. Navy's Naval Air Warfare Center Training Systems Division (NAWC TSD). Her efforts with the Navy focused on high risk research and development aimed at rapid transition of virtual communications capabilities to the Field and Fleet. She earned her Ph.D from the Industrial Engineering and Management Systems department of the University of Central Florida.

# Predicting Business Process Performance with 'Real World' Queuing

**Joseph S. DeKeyrel**
**Raytheon Technical Services Company**
**Orlando, FL**
**Joseph_S_DeKeyrel@Raytheon.com**

**Stephanie J. Lackey**
**Institute for Simulation and Training**
**Orlando, FL**
**SLackey@ist.ucf.edu**

## INTRODUCTION

Procurement is a challenging task. Material procurement can be straight forward, if the material is well defined, mature, and available. The procurement of systems can certainly be more challenging – usually requiring the creation of detailed specifications and performance parameters for execution within some concept of operations. Similarly, the procurement of services can be challenging – requiring the creation of detailed statements of work, again in the context of a concept of operations. Further complicating procurement in the services realm is the seemingly non-deterministic behavior of the primary service components – human beings. For the purposes of this paper, the discussion will focus on the procurement of services, and more specifically on the opening full stanza of that process – from agency initiation through provider submittal of a proposal.

In the somewhat trivial case of a single, isolated procurement – the remainder of the material provided will offer little value. However, in the much more common situation of multiple procurement actions being prosecuted simultaneously amongst a procuring agency and its attendant industrial base, the ability to make reasonable planning predictions *based on the resource constrained behavior* of the aforementioned aggregated organizations and specifically the length of time necessary to bring a given procurement to fruition is quite valuable.

These planning predictions – if accurate enough – provide the basis for backwards planning from required service commencement date, back through transition, back through source selection and award, back through bid preparation, to the release of the Request For Proposal (RFP) and perhaps even further back to a Sources Sought or Broad Agency Announcement (BAA). As an additional benefit, the aggregation of these predictions would allow for the assessment of the relative resource impacts of various procurement strategies, e.g. a single large procurement versus several smaller procurements.

Accurate determination of due dates for the delivery of custom services based on non-technical specifications is a challenging task. Competition and struggling economies conspire to drive limited fixed staffing levels to control costs which is at odds with having sufficient resources necessary to quote these due dates in a timely fashion. An environment that is extremely contentious with respect to the necessary resources and offering little in the way of consistent prioritization only exacerbates the situation. And finally, when consumers demand both demonstrably strict dates (shortening their execution cycles) and suffer severe penalties for exceeding those dates (failure to meet required service deliveries) the situation becomes nearly untenable. The proposed solution describes an artful combination of automated analysis and efficient, embedded simulation that might be successful in resolving this stark situation.

### Prerequisites

In order to apply the methodology described here, a practitioner should already have (1) developed a functional, transaction-based workflow system, (2) performed an initial, manual data analysis of the processing times, queuing behavior and rework rates, and (3) built a representative discrete event simulation (DES) model of the workflow process to validate understanding of the practitioner's system.

However, prior to any of the above prerequisites being satisfied for a government procurement model several hurdles must be overcome. The first, and perhaps most counter-intuitive is to *expand the scope* of the system under test such that it includes not only the

governmental processes, but also the processes executed by the industrial base. These various supplier processes subsume much of the variability in the overall process on a step-by-step basis, but when magnified by multiple suppliers executing their processes in both series and parallel, the overall impact is daunting. But without considering these impacts, the interested practitioner would be doomed to under estimate the overall variability in the system and likely end up with little confidence in the predictions so generated. So, to overcome this fatal first misstep, the process to be modeled must include the procuring agency and at least the first tier of suppliers. With this process definition settled, a workflow system that supports data collection on this process can be constructed.

A skeptical reader might scoff at the notion that suppliers would be willing to allow a view into the performance of these sensitive processes. Gleaning this information is, in fact, the second hurdle to overcome, but based on the authors' observations it can be accomplished when the suppliers are sufficiently motivated to participate.

**Similarity of Processes**

In the primary author's business context, the business process can be depicted as shown in Figure 1. The process steps and data (except those with a grey background) are captured in a single workflow system.

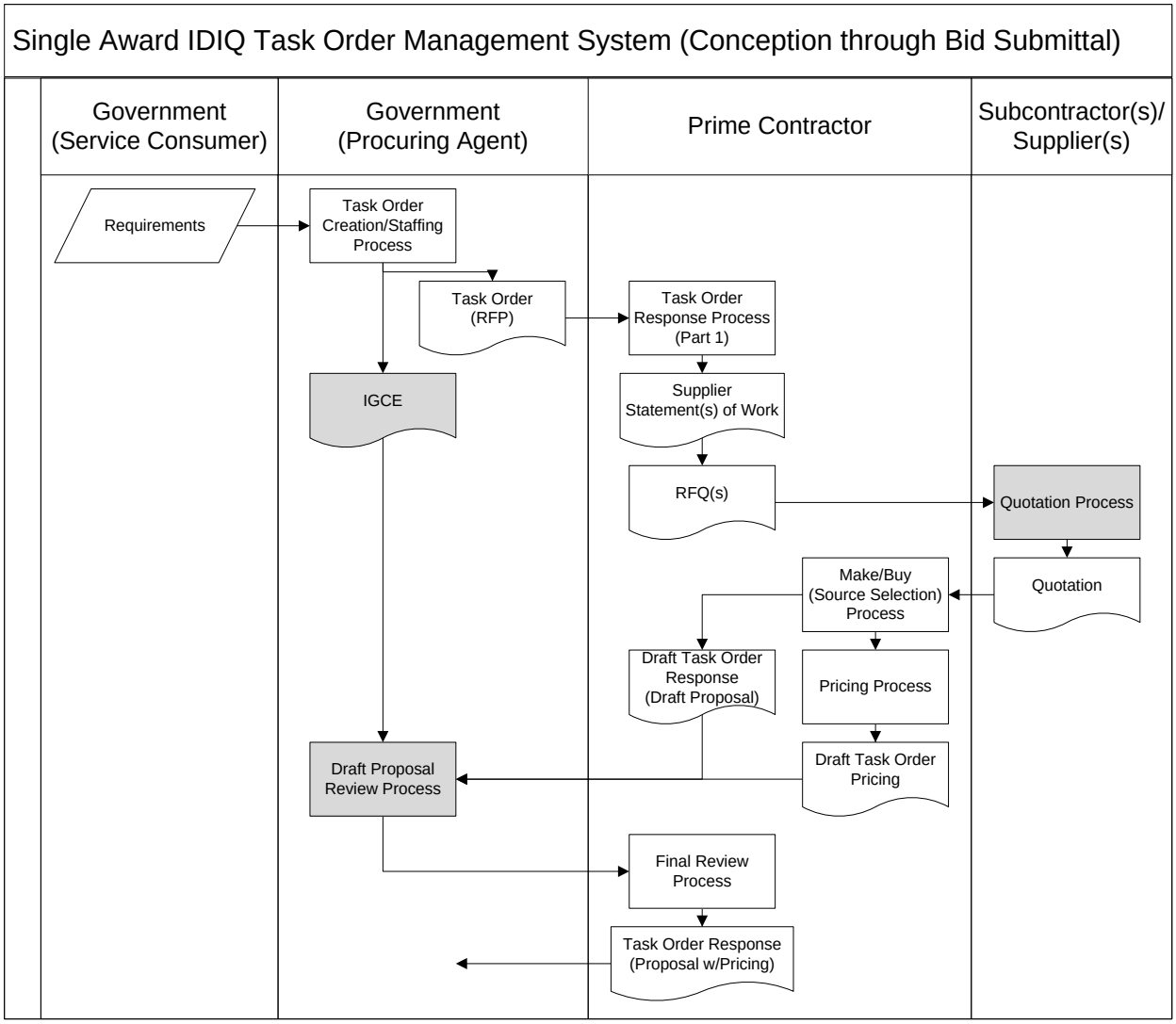


**Figure 1 - Single Award Process**

Next consider a corresponding process as it might be implemented in a Multiple Award context as shown in Figure 2. As in the previous diagram, the items in grey would likely be processed outside of the workflow system's purview.

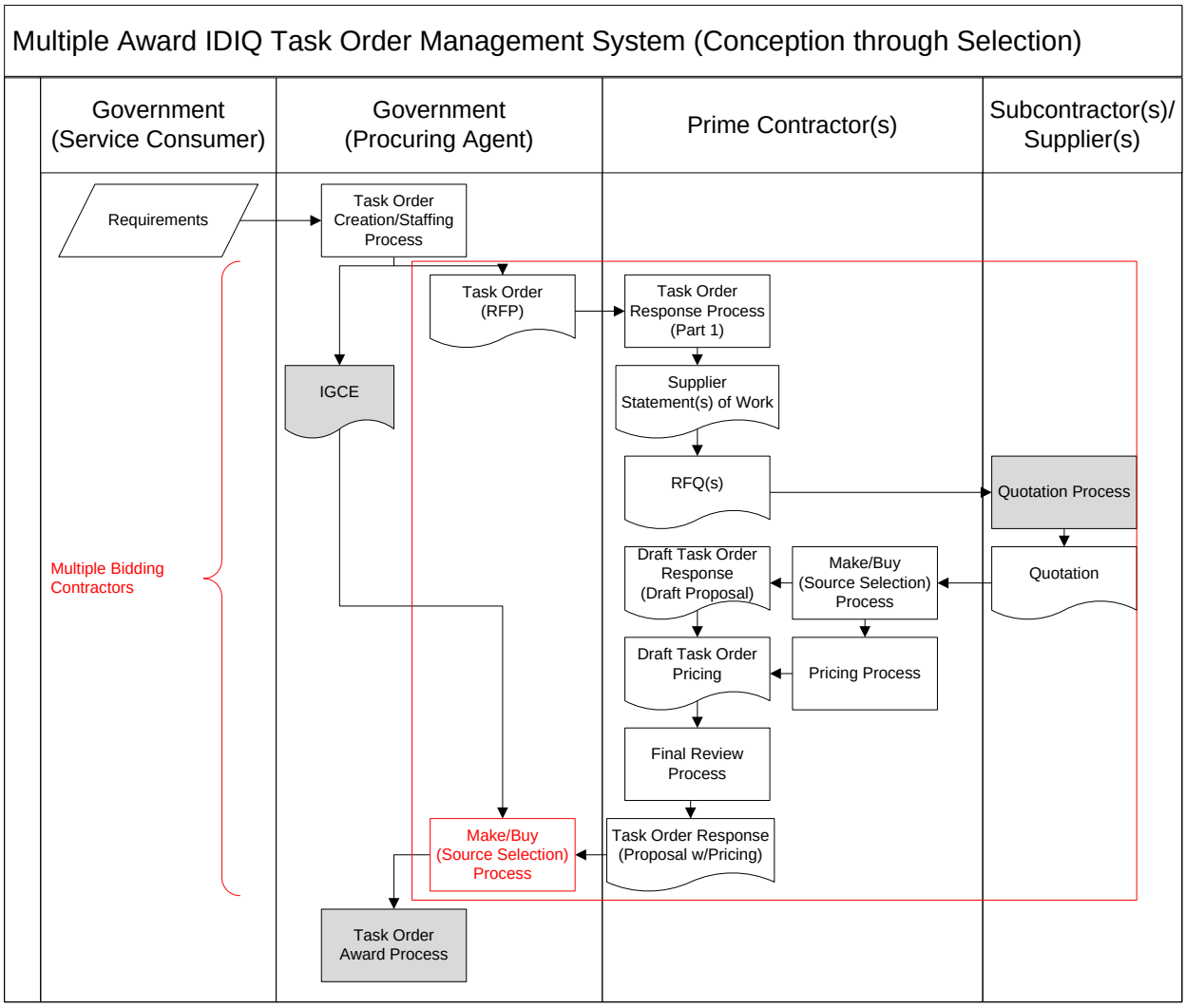


**Figure 2 - Multiple Award Process**

An examination of the portion of the multiple award process contained within the red outline (from Task Order RFP through Make/Buy Process) in relation to the single award process readily supports the conclusion that methods applicable to the single award process should similarly apply to the multiple award system. With this relationship in hand, the authors' will relate the research performed on the single award process and, at the conclusion, apply those results to the multiple award process.

**Scope of Problem**

Before reviewing the relevant literature, it will be worthwhile to highlight the underlying problem. As will be shown later, the model of a workflow system which might implement a business process such as those shown here can likely be decomposed into a network of servers with queues. Such formulations normally require assumptions about how often items arrive to be served, the sequence of servers that items will visit, the amount of time necessary for processing items at the servers, and perhaps some rules about how the servers manage the queues of items waiting to be processed. Such systems, at least when not idle, tend to be dominated by the amount of time that items spend waiting to be served as opposed to the amount time that they are processed. Conceptually then, one might

estimate when a given item would exit the system by taking its arrival time and adding both the total amount of processing time required, and the total amount of waiting time required. If reasonably accurate assumptions about the processing times may be made, and the servers always process the items in a first come, first served basis then there are many techniques that will accurately estimate the waiting time – problem solved! In the authors' experience, however, when human actors (especially knowledge workers) perform as the servers, the first come, first served behavior may not be a valid assumption. In this case the waiting time can only be expressed as a non-deterministic relationship between the arrival rate, the number of items in the system, their collective processing times, and some descriptions of the queue handling behavior and other routing information. A more detailed mathematical treatment of this topic can be found in previously published work (J. DeKeyrel, Geiger, Malone, Lackey, & Mollaghasemi, 2011).

The following relationship summarizes the salient difficulty in predicting turn-around times (TATs) in a system with non-standard queuing behavior.

$$\boldsymbol{W_i \equiv f\left(IAT, \vec{P}, \vec{Q}, \vec{R}\right)}. \qquad (1)$$

Where $W_i$ is defined as the total time spent waiting by item *i* and IAT is the inter-arrival time for items that appear after item *i* arrives, and $\vec{P}$, $\vec{Q}$, and $\vec{R}$ are the vectors of processing times, queuing behaviors, and rework rates respectively for the other items in the system. Note that the arrival process need not be stationary, and in fact, is not in the subject system (J. S. DeKeyrel, 2010).

The authors' proposed solution to determining $W_i$ is then to (1) construct an embedded DES model, (2) determine the parameters for that model applicable at the point in time where job *i* enters the system, (3) determine the properties of job *i* necessary for representation within the model, and (4) to repeatedly execute the model until an acceptable margin of error on predicting its time in system can be achieved.

## RELEVANT LITERATURE

The following sections highlight some of the salient literature that bears upon this topic from the areas of due date quoting, predictive use of models, and embedded modeling.

### Due Date Quoting

Cheng and Gupta (Cheng & Gupta, 1989) produced a survey of the existing research with respect to due date determination. In this survey, Cheng and Gupta open by pointing out that meeting due dates is extremely important to practicing managers. They then utilize a classification scheme first proposed by Elion (Eilon, 1978) which has six (6) dimensions: (1) Static versus Dynamic, (2) Deterministic versus Stochastic, (3) Single-product versus Multi-product, (4) Single-processor versus Multi-processor, (5) Theoretical versus Practical, and (6) Exogenous due dates versus Endogenous due dates. Since exogenous due-dates obviate due-date quoting and lead directly to sequencing and scheduling problems, Cheng and Gupta focus their attention on endogenous due-dates. Using the above classification scheme they conclude that there is very little extant research on Dynamic, Complex, Multi-processor systems. And after noting that better predictors would be beneficial, if practical, they conclude that there is a need for more practical and applied research in this area.

Alfieri (Alfieri, 2007) proposes two new quoting policies based on setting a static Safety Time (ST) parameter and noting that setting this parameter dynamically could be time consuming. The performance of these quoting policies, which both presuppose a First-Come-First-Served (FCFS) ordering, is compared to the Total Work Content (TWK) policy when jobs are sequenced by Shortest Processing Time (SPT), Earliest Due Date (EDD) and First-In-First-Out (FIFO). These comparisons are predicated on batch scheduling (ignoring subsequent arrivals), deterministic processing times and non-permutation sequencing. With these simplifications, her results indicate that TWK outperforms both of her proposed policies. She notes that estimating flow times for more complicated systems is a suitable topic for future research.

Subsequent to the survey conducted with Gupta discussed above, Cheng (Cheng, 1991) describes an efficient and optimal sequencing algorithm when using the slack due date quoting policy. Cheng simplifies the system under consideration by assuming that once a set of jobs is sequenced, no subsequent jobs will affect the systems performance, there will be no re-sequencing of the jobs between stations and all of the earliness and tardiness costs are constant. In effect, the lack of consideration of arrivals and non-permutation scheduling becomes a presupposition of FCFS. In this scenario Cheng concludes that an SPT sequence is

optimal although this conclusion is at odds with the findings of Duenyas and Hopp below.

Duenyas and Hopp (Duenyas & Hopp, 1995) propose an analytical framework for evaluation of various job sequencing rules given that flow times can be optimally predicted. Working through a series of increasingly generalized scenarios they conclude that an EDD sequence is optimal if the tardiness penalty is constant for all customers and proportional to the tardiness which seems to contradict Cheng (Cheng, 1991) above. To achieve this result Duenyas and Hopp only assume that preemption does not take place. The result of an EDD sequence being optimal is useful in that it provides direction for redesigning the workflow system in the authors' construct to encourage EDD processing order but is not helpful in determining the optimal due dates.

Similar to Duenyas and Hopp above, Lawrence (Lawrence, 1995) presupposes that the practitioner either has a simple system with closed-form flow time estimates, or has some way to determine flow times for complex systems. With that as a precondition, he describes an analytical approach to setting due dates based on previously observed forecasting errors. While Lawrence proposes to fit the forecasting errors, which he refers to as "G", using a Ramberg-Schmeiser distribution, he concludes that Erlang and Gaussian distributions worked equally well in his research. Lawrence makes three observations that are particularly germane in this context: (1) exponential smoothing of the forecasting error distribution parameters enhances the accuracy of the fit, especially in time-dynamic situations, (2) various measures of performance lead to differing uses of the error distribution, e.g. Mean Absolute Lateness is minimized by adding the median of the error distribution to the predicted flow time, Mean Square Lateness is minimized by adding the mean of the distribution to the predicted flow time , and service level matching is met by adding the target percentile of the distribution to the predicted flow time, e.g. $G^{-1}(0.9)$ for a 90% Service Level, and (3) the analytic due date quoting policies that include information about the current system state outperform those that do not at least in the simple scenarios that the author evaluates specifically. Additionally, Lawrence's paper provides a good summary of the most common analytic quoting policies which will be useful for comparison with the proposed modeling-based approach presented herein.

Van Ooijen and Bertrand (van Ooijen & Bertrand, 2001) introduce a distinction in terminology intended to allow some leeway between the tightly estimated Internal Due Date (IDD) and the slightly looser External Due Date (XDD). To set this difference, which is analogous to $e_i$ in the problem description from section 1.2, or the Safety Time from Alfieri, or Lawrence's error distribution, G, the authors propose to adjust the XDD using the ratio of the current level of work in progress (acwip) to the average level of work in progress (nwip). Using variations of this quoting policy various sequencing rules were applied and the optimal cost per order was established over a variety of relative earliness/tardiness combinations. Van Ooijen and Bertrand's results bring some closure to the disagreement between Cheng (Cheng, 1991) and Duenyas (Duenyas & Hopp, 1995) by noting that when earliness and lateness penalties are of similar magnitude then SPT sequencing works best; however, when tardiness penalties are much larger than earliness costs a due date sequencing rule is best. Another interesting observation that can be made from the data is that in spite of the dependence on FCFS sequencing in much of the literature, FCFS provided among the worst actual performance of the sequencing rules tested – it does however provide the best predictions of performance.

Rajasekera, Murr, et al. (Rajasekera, Murr, & So, 1991) open by observing that including more information into the dynamic flow time prediction process produces better results. Much of the paper subsequently focuses on an analytical description of a load-balancing algorithm that could be implemented in an information system integrated with the manufacturing system. The authors conclude that after applying their load balancing procedure and assuming FCFS processing, then setting due-dates is straightforward even when taking into account the jobs already in the system. As a parting note, the authors concede that more complex work centers would require more complex queuing decomposition methods and further analysis.

**Predictive use of DES Modeling**

Much of the existing literature talks about using models of systems to conduct experiments where the objective is to optimize system performance by adjusting resources or queuing behavior (Kelton, Sadowski, & T.Sturrock, 2004; Law & Kelton, 2000).

There is some literature that seeks to use the model to evaluate differing courses of action such as selecting a sequence of jobs to be scheduled. For example, Azzaro-Pantel, Bernal-Haro et al. describe using a

combination of discrete event simulation and a genetic algorithm to optimally dispatch tasks in a job shop environment, with the genetic algorithm generating the sequences and the DES model evaluating each sequence (Azzaro-Pantel, Bernal-Haro, Baudet, Domenech, & Pibouleau, 1998). In a related fashion, Reijers discusses using short-term simulations coupled with work flow to provide decision support, i.e. scheduling additional resources during peak loads (Reijers, 1999). Much less of the literature discusses the potential for use of the faithful model to make predictions about the system *just the way it is*. Rojanapibul and Pichitlamken make some excellent observations about using embedded simulations to calculate prediction intervals in a flow shop environment (Rojanapibul & Pichitlamken, 2005). Cates and Mollaghasemi describe the use of simulation to predict project completion dates and thereby enhance visibility of risk to better manage completion of complex projects (Cates & Mollaghasemi, 2007). In both of these cases, though, the job parameters were reasonably established before predictions were made.

## DEVELOPMENTAL DETAILS

As called out in the prerequisites, and based on the process in Figure 1 as well as the theoretical underpinnings of discrete event simulation (Law & Kelton, 2000), the primary author constructed an automated workflow system to support this process. The workflow system was then mapped into a discrete event simulation model which is depicted in Figure 3.

The authors' prototypical solution for implementing this methodology is composed of two distinct, but closely inter-related components. The first component, which replicates the previously mentioned manual analysis as an automated process, uses historical data to determine descriptive parameters. The second component is an embedded simulation model that makes use of these descriptive parameters to replicate the behavior of the target system. It is important to note that the predictive power of this construct is dependent on both components, which must act in concert.

### Automated Analysis

The automated analysis component performs five major functions: (1) decompose the departure transactions (by job and by station) from the workflow system into Departure and Arrival events, (2) using the correlated Departure and Arrival events determine the rework rate of the sample of jobs by station, (3) using the correlated events by station, determine the queuing behavior for that station, (4) using the correlated events by station, decompose the total time at a station for a job into waiting time and processing time and fit the processing times to a valid statistical distribution, and (5) utilizing the transaction logs, determine the inter-arrival rate per month. The last four functions output their results as a series of parameters to be used by the embedded simulation.

The first function is a pre-processing step facilitating the remaining functions. As mentioned, the system in question is an electronic workflow system. As such, there is no perceptible transportation delay. Without transportation delay, the decomposition of the departure transactions simply requires the creation of a departure event from the current station, and an arrival event at the next station visited by the job. The times of occurrence for each of these events are identical; the only complicated aspect is determining the next station visited. As this complication is purely self-inflicted by the authors' implementation of transactions, recording the details of overcoming this particular hurdle will be glossed over. A sage practitioner would be well served to capture both the source and destination stations within the departure transaction and thus avoid this step entirely. As the output of this step is only used as the input for the subsequent three steps, there is no need to store these results back to the database.

The second function uses the correlated departure and arrival events created by the first function to determine rework rates. This is accomplished simply by implementing a two-level, nested-case construct which takes at the outer-level the source station, and at the inner-level the destination station. The rework status per job is then captured as a logical action, in the authors' case a job is accepted, rejected or returned without further action. The relative frequencies of these actions are recorded by station as model parameters in the database and are used by the branch components to correctly route jobs from one station to the next – this pairing of analytical and simulation components directly addresses $\vec{R}$ from Equation 1.

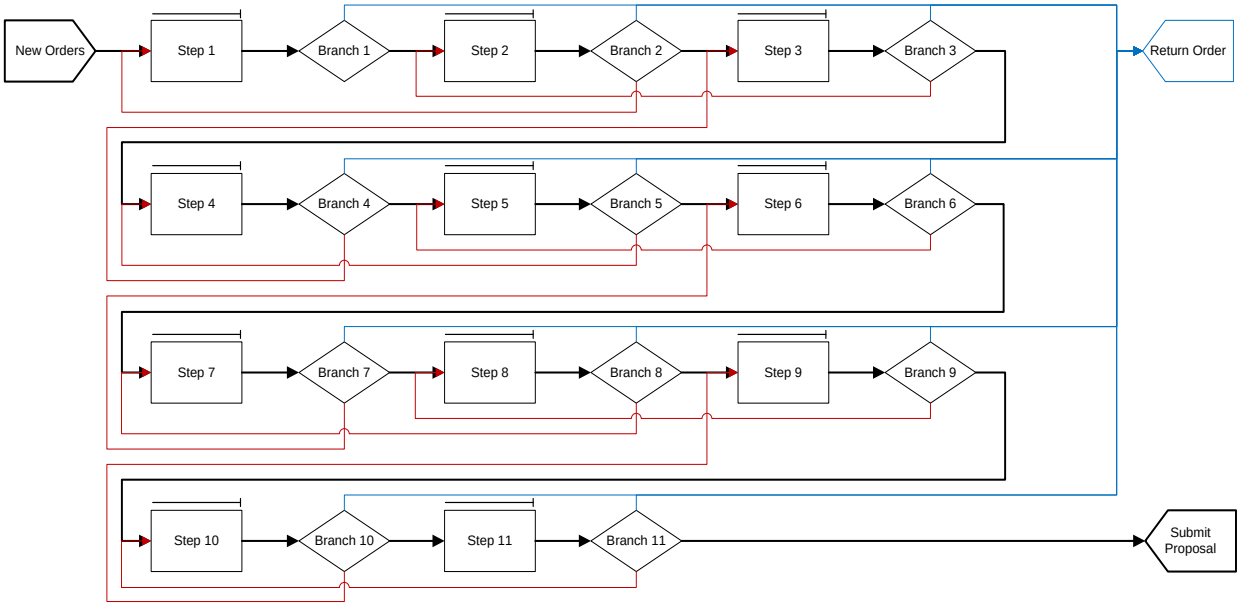


**Figure 3 - Discrete Event Simulation Model**

The third function, determining the queuing behavior, is considerably more interesting to describe, and is in fact, half of the novel aspect of the authors' formulation for attacking $\vec{Q}$ in Equation 1. In general terms, the concept of the function is similar to executing a DES in reverse. In a normal DES, both the processing time for a job, and the queuing policy for a station are specified and the result for the job is the departure time from the station (Law & Kelton, 2000). In this case, however, the arrival and departure times are known and the results of the analysis are the processing time for the job, and the queuing behavior of the station. More specifically, the historical jobs arriving at a given station are processed in time-order of their arrival at the station but the jobs are placed in the queue based on their, known a priori, departure time. Executing this process one input job at a time, it is possible to determine the queue insertion location at the station, and the accumulated processing time for the job.

As an example of this process consider the following sequence: job 1, which arrives at station X at time 0 and is known to have departed at time 20, finds station X empty and idle; since the server is empty and idle, job 1 is immediately placed in service (location = 0, queue depth = 0) and begins to accumulate processing time. Job 2 (arrives at time = 5, will depart at time = 21) arrives at station X; since the station is not idle the departure time of the newly arrived job is compared to that of the job in service; since job 2 will depart after job 1, it is placed in queue; since the queue is empty, job 2 is queued at location = 1, queue depth =1. Job 3 (arrives at time = 10, will depart at time = 30) arrives at station X; since job1 is still in service, departure times for jobs 1 and 3 are compared; job 3 will depart after job 1, so job 3 is queued; since job 3 will depart after job 2, it is queued after job 2 at location 2 and queue depth = 2. Job 4 (arrives at time = 15, will depart at time = 25) arrives at station X; since its departure time is after job 1 (still in service), job 4 will be queued; since job 4 will depart after job 2 and before job 3, it is queued at location = 2, queue depth = 3 which is recorded as '2/3'. Executing this scenario, and stopping at time = 15 is represented graphically in Figure 4.

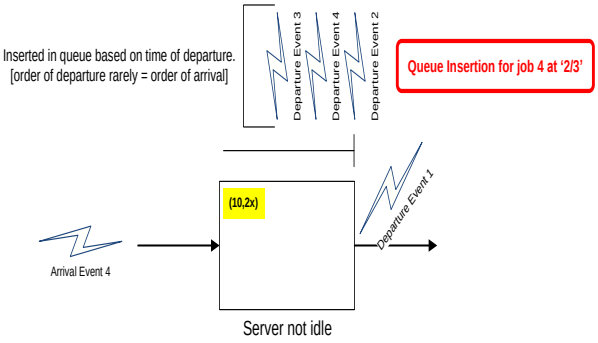


**Figure 4 - Queuing Behavior Determination**

The output of this function is three parameters per station specifying the fraction of jobs that preempt, queue at the head-of-line, and queue at the tail-of-line. Jobs that don't meet any of the three criteria are

assumed to be randomly placed in the queue between head-of-line and tail-of-line.

The fourth function separates the processing time from the waiting time and then fits the processing times to a statistical distribution (Law & Kelton, 2000). This statistical distribution addresses, in conjunction with the server simulation component, the $\vec{P}$ component from Equation 2. In the authors' implementation, the first portion of this function – separating processing and waiting times for a job at a station – is accomplished by a combination of the virtual server methods. Interested readers may wish to review the more detailed description of these processes available in (J. DeKeyrel et al., 2011).

The second portion of the function uses a well known formulation to convolve the resulting processing times at a given station such that a linear, least-squares regression of the convolved data exhibits the shape and scale parameters of a Weibull distribution fitted to the unprocessed data (Law & Kelton, 2000). Similar to the implementation(s) above, the newly calculated parameters are combined using exponential smoothing – as in the second and third functions – with the existing parameter values and the resultant, smoothed values stored back into the database, two parameters per station. Unlike the previous implementations above, however, a Kolmogorov-Smirnov goodness of fit test is executed between the source data and the fitted distribution, and the newly calculated parameters are only combined with the existing parameters if the test statistic is less than the adjusted critical value for the sample size (Law & Kelton, 2000).

As the reader may have already surmised, the fifth function, calculating the inter-arrival rates by month, when coupled with the source component of the simulation, completes the input parameters to Equation 1, namely IAT. This function is executed very simply using an SQL query which aggregates the arrivals by month for the previous 12 months. The more interesting aspects of this function reside in the simulation component discussed below.

### Embedded Simulation

To implement the embedded simulation portion of the solution, the authors added to the JSim library to introduce a multi-way branch. Other components were extended to implement novel behaviors such as a server that allows for preemption, and a source that allows for a non-stationary arrival process.

## SYSTEM UNDER TEST

The model of the system under test is implemented as a top-level simulation object. This object has one source component implementing non-stationary arrivals as indicated above and containing an order factory producing orders in accordance with the processing time distributions based on the Weibull $\vec{P}$ parameters, including the special "target" order. The simulation object instantiates 11 servers which, in conjunction with their attendant queues, implement empirical queuing behavior in accordance with the $\vec{Q}$ parameters from the analytical component. It also instantiates 11 branches (3-way) that implement rework based upon the $\vec{R}$ parameters. Finally, the simulation implements two sink components, one for capturing objects successfully traversing the system and a second for objects that are returned to the customer without further action.

These components are instantiated, logically connected as pictured in Figure 3, initialized with the parameters as mentioned above, the queues pre-loaded with jobs according to the current date's queues. At this point the target job is introduced to the system, and the simulation clock started. The simulation run terminates when the target job exits via the first sink.

To facilitate statistical analysis, the target jobs from each replication of the simulation are maintained until the desired number of replications has been executed. At that point the collection of target jobs can be summarized, in this case by determining the upper confidence level for the mean of the turn-around time.

## TEST METHODOLOGY

As the actual system under test is, in fact, a transactional workflow system, it is possible to roll the systems state back to any point in time covered by the transaction log. Utilizing this capability it is possible to (1) determine actual turn-around times for jobs entering the system on any given day, and (2) to execute both the analytical and simulation components against the data that was available on that same day. With both data sets available simultaneously it is possible to compare the actual and predicted data side-by-side.

The actual turn-around times were gleaned from the workflow system through an SQL query of the database that provides persistence to the workflow system. This query was structured such that the output consisted of the date, the mean turn-around time of the

jobs that entered the system on that date, and the number of jobs entering on that date. Using this data it was then a simple bit of manipulation in Microsoft Excel to generate a time-weighted average turn-around time looking back 10 days to smooth the necessarily jagged plot of mean turn-around times.

The predicted turn-around times were generated by providing a "main" function that specified a date for simulation such that the analytical component could execute as if it were that date and looking 180 days into the past to calculate the simulation parameters, and then using that same date, the simulation component could execute 200 replications of the model capturing the upper confidence limit (UCL) of the mean turn-around time. After the analytical and simulation components had executed for the date specified, the date was incremented by 1 and the process repeated until the desired end date was reached. The output of the components was adjusted such that the output was the date, the number of jobs in queue on that date, and the UCL of the time in system for a new job on that date. With the two data sets described it is a simple matter to match the actual data and the predicted data by date, again using Excel.

## RESULTS

Initial results of the tests conducted indicate an expected result – that the turn-around time predicted for a given job is closely correlated to the number of jobs in queue when the new job enters the system as shown in Figure 5. The red line in the figure represents the 90% UCL for the mean turn-around time predicted by the model, while the blue line – plotted against the secondary y-axis – represents the total number of jobs in the system when the target job arrives.

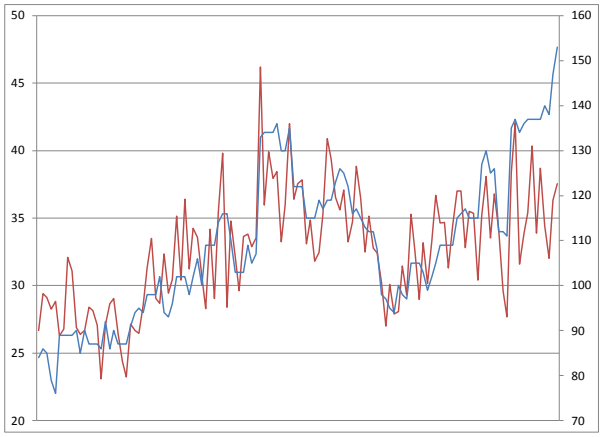


**Figure 5 - Turn Around versus Items in Process**

Of more practical benefit is the indication of a good correlation between the predicted turn-around times for a given day, and the actual, observed turn-around times for jobs entered on that day as shown in Figure 6. The red line is the same as in Figure 5 – the 90% UCL for the mean, but the green line represents the mean turn-around time for the actual jobs that entered the system on that day.

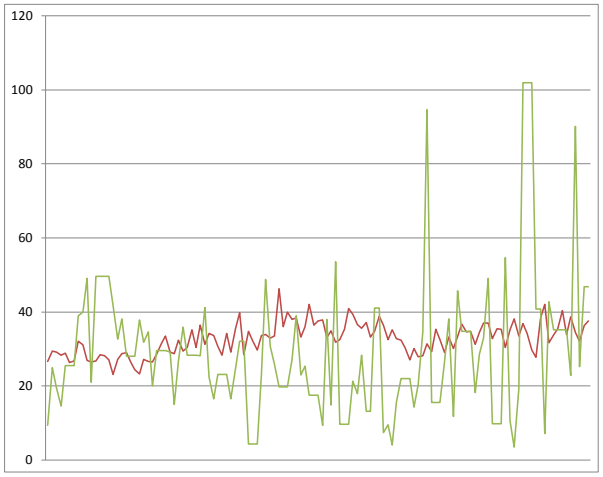


**Figure 6 - Predicted versus Actual Turn Around**

The performance indicated in Figure 6 above is actually quite good. Simply using the UCL of the mean flow time for predicting the due dates yields a service level of approximately 65%. Adjusting the flow time by adding in some multiple of the variance of the forecasting error $e_i$ ($1.285\sigma_e$) allows the achievement of a 90% service level. Using this implementation of Lawrence's methodology (Lawrence, 1995) achieved 92% during the historical period analyzed.

And while achieving at least a 90% score is desirable for the process owner, it may be more attractive to a customer to tune the predictive subsystem for an 80% service target and incentivize the process owner to achieve the next 10%. An interesting side benefit of this methodology is that it provides a ready mechanism for continuous improvement, i.e. if the processor is successful in achieving 90% during this period, future job flow times will be based on this tighter standard.

## CONCLUSIONS

The authors' previous work indicated that the existing, deterministic methods of quoting due dates suffered when applied to systems not based on FCFS queuing and argued that investigation of a stochastic approach

was warranted. This paper documents that investigation, and indicates that a carefully crafted mix of automated analytics and embedded simulation might indeed provide a practical alternative for higher-fidelity due date quoting in systems with non-standard queuing behavior and high levels of rework.

Returning to the opening premise, the authors assert that the presented methodology could be implemented across the extended procurement enterprise as depicted in Figure 2, and allow the government to more accurately plan for, and schedule, procurement activities. This would necessitate the definition of a broad-scoped business process including the government agency and its industry partners, the development of a ubiquitous workflow tool to be used by all parties, and the fitting of a discrete event simulation to that process.

## ACKNOWLEDGMENTS

The authors gratefully acknowledge the continuing support of Raytheon for this research.

## REFERENCES

Alfieri, A. (2007). Due date quoting and scheduling interaction in production lines. *International Journal of Computer Integrated Manufacturing, 20*(6), 579-587.

Azzaro-Pantel, C., Bernal-Haro, L., Baudet, P., Domenech, S., & Pibouleau, L. (1998). A two-stage methodology for short-term batch plant scheduling: discrete-event simulation and genetic algorithm. *Computers and Chemical Engineering, 22*(10), 1461-1481.

Cates, G. R., & Mollaghasemi, M. (2007). The project assessment by simulation technique. *EMJ - Engineering Management Journal, 19*(4), 3-10.

Cheng, T. C. E. (1991). Optimal assignment of slack due-dates and sequencing of jobs with random processing times on a single machine. *European Journal of Operational Research, 51*(3), 348-353.

Cheng, T. C. E., & Gupta, M. C. (1989). Survey of scheduling research involving due date determination decisions. *European Journal of Operational Research, 38*(2), 156-166.

DeKeyrel, J., Geiger, C., Malone, L., Lackey, S., & Mollaghasemi, M. (2011). *Real-Time Assignment of Due Dates within Workflow Management Systems.* Paper presented at the (pre-press) Industrial Engineering Research Conference, Reno, NV.

DeKeyrel, J. S. (2010, 8-10 November 2010). *Processing predictions through embedded simulation.* Paper presented at the Proceedings of the IASTED International Conference, Software Engineering and Applications (SEA 2010), Marina del Rey, CA.

Duenyas, I., & Hopp, W. J. (1995). Quoting customer lead times. *Management Science, 41*(1), 43-43.

Eilon, S. (1978). *Production scheduling.* Paper presented at the Operational Research '78: Eighth IFORS International Conference on Operational Research, North-Holland, Amsterdam.

Kelton, W. D., Sadowski, R. P., & T.Sturrock, D. (2004). *Simulation with Arena* (3rd ed. ed.). New York: McGraw Hill.

Law, A. M., & Kelton, W. D. (2000). *Simulation Modeling and Analysis* (3rd ed. ed.): McGraw-Hill.

Lawrence, S. R. (1995). Estimating flowtimes and setting due-dates in complex production systems. *IIE Transactions (Institute of Industrial Engineers), 27*(5), 657-668.

Rajasekera, J. R., Murr, M. R., & So, K. C. (1991). A due-date assignment model for a flow shop with application in a lightguide cable shop. *Journal of Manufacturing Systems, 10*(1), 1-7.

Reijers, H. A., van der Aalst, W.M.P. (1999). *Short-Term Simulation: Bridging the Gap between Operational Control and Strategic Decision Making.* Paper presented at the IASTED International Conference - Modeling and Simulation (MS '99), Philadelphia, Pennsylvania - USA.

Rojanapibul, K., & Pichitlamken, J. (2005). *Assessing risk in a job schedule: Integrating a scheduling heuristic and a simulation model to a spreadsheet*, Orlando, FL, United states.

van Ooijen, H. P. G., & Bertrand, J. W. M. (2001). Economic due-date setting in job-shops based on routing and workload dependent flow time distribution functions. *International Journal of Production Economics, 74*(1-3), 261-268.