

Descartes-Origins: A Simulation-Based Approach to Costs and Schedules of Future Spaceplane Development Programs

Michael J. Kelly¹, Ronald P. Menich², and John R. Olds³
SpaceWorks Engineering, Inc. (SEI), Atlanta, GA, 30338

Cost and schedule estimating for large aerospace projects has long been considered a difficult problem. Many industry standard tools are based primarily on historical analogy databases along with inflation tables. This prevents most estimates from accounting for interaction between resources on a larger scale, the process flows involved in design, development, testing, and production, and the variability that is naturally generated as a result of these processes.

SpaceWorks Engineering, Inc. (SEI) is developing, through research and quantitative modeling, a discrete event simulation to predict the time and resources required to design, develop, test, and produce future fleets of reusable launch vehicles. The simulation model is being built with spaceplane-type architectures in mind, and will allow for rocket, turbine, combined cycle, or other forms of propulsion. This modeling goes beyond traditional cost estimating by focusing on the processes inherent in design and production, and how those processes utilize resources.

The model, named Descartes-Origins, is implemented in the discrete event simulation software Arena, commercially available from Rockwell Automation. Descartes-Origins utilizes similar methodologies to those used in Descartes-Hyperport, SEI's successful ground operations modeling tool. The model will take as inputs cost estimates generated by industry-standard cost models, along with answers to questions about management strategies, quantities of contractors, and budgetary requirements. Outputs will be focused on distributions of time and cost projections, divided amongst appropriate program phases. Output estimates from this model will be multi-dimensional, including ideas of how a project might respond to changes in budget, and where schedule slippages and cost overruns are likely to occur.

In the early part of the twenty-first century, both the theoretical underpinnings and actual simulation software capabilities are present to create a usable, activity-based systems analysis capability for metrics related to reusable launch vehicle development and production.

Nomenclature

DDT&E = Design, Development, Test, & Evaluation	NRA = NASA Research Announcement
DES = Discrete Event Simulation	SEI = SpaceWorks Engineering, Inc.
I-RaCM = Integrated Risk and Cost Model	TFU = Theoretical First Unit
LCC = Life Cycle Cost	VB = Visual Basic

I. Introduction

IN the not-too-distant future, it is extremely likely that the aerospace industry will design and build another generation of reusable space launch vehicles (RLVs). As the amount of space-bound traffic continues to increase over the next few decades, reusable vehicles have the potential to offer significant cost savings over existing or new

¹ Operations Engineer, Engineering Economics Group, 1200 Ashwood Parkway - Ste. 506, AIAA Member.

² Space Scenario Analyst, Engineering Economics Group, 1200 Ashwood Parkway - Ste. 506, AIAA Member.

³ Principal Engineer, 1200 Ashwood Parkway - Ste. 506, AIAA Associate Fellow.

expendable launch systems. Reusability offers obvious advantages in total fleet acquisition costs for any extended program, while the challenge of maintaining this advantage lies in designing vehicles that are cheap and efficient to operate. A well-designed RLV should be able to be turned around from landing to launch quickly and efficiently enough to support increased demand for launches. While this should result in savings over time, there is the risk of potentially higher development and per-vehicle production costs when compared to an expendable vehicle. For a complete picture, the total Life Cycle Cost (LCC) of a program has to be calculated as the sum of initial Design, Development, Test and Evaluation (DDT&E), Theoretical First Unit (TFU) production, additional production, and ongoing (or recurring) operations costs (Facilities and Cost of Failure are part of LCC too – should we include that here?). Along with the cost estimates themselves, it is important to understand how costs are spread over the life of a program, from concept inception through the completion operational plan.

Historically, many of these analysis areas are approached independently. Often DDT&E and TFU are estimated using a single tool, while recurring costs are an independent calculation. (Add facilities / cost of failure here if you choose to include it) Cost-spreading may be done simply, with low-fidelity estimates of metrics like ‘development time’ and costs roughly spread out along that time period (beta distributions etc...). Additionally, the various components of cost are often estimated using different methodologies and various underlying assumptions, which can lead to further inconsistencies and less precise estimates. In the last few years, various efforts have been underway to improve the understanding of cost and schedule interactions. Since 2008, SpaceWorks Engineering, Inc. (SEI) has conducted a study effort to develop new tools that use consistent methodologies to obtain estimates of life cycle costs and schedules. When applied together, these tools can be used to give program managers and research teams a more complete and accurate picture of the financial consequences of their vehicle design decisions.

A. Discrete Event Simulation

The methodology being used for SEI’s tools is called Discrete Event Simulation (DES), also known as the event scheduling approach. DES is a technique developed in the industrial and systems engineering community that has been applied in industries as varied as healthcare, banking, manufacturing, shipping, and commercial aviation. In general terms, DES allows any complex set of processes to be described in terms of events occurring at discrete points in time. A non-instantaneous event is re-defined by instantaneous start and end points. A DES model keeps track of all the events that are scheduled to occur. As time moves forward, when an event is reached on the schedule, its effects on the system (e.g. new resources becoming available, an inspection being completed, etc.) are recorded, sometimes new events are added to the schedule, and the simulation moves on to the next event. This is very computationally efficient, since it only requires a single event to be processed at a time, and it enables very complex systems to be analyzed by breaking them down into discrete events and placing all the events on a single schedule. DES can be a powerful tool for many kinds of systems analysis, and is ideally suited to problems like production and assembly, or ongoing ground operations.

Basic DES models can be built in spreadsheets, but larger models generally employ DES-specific software. Rockwell Automation’s Arena software, built on the DES language SIMAN, is one of the industry leaders and was selected by SEI for this study. Arena’s Basic Edition provides a ‘Basic Process Panel,’ which allows users to build models with 8 types of modules, or ‘blocks’: create, dispose, process, decide, batch, separate, assign, and record. The software also gives spreadsheet-based control over resources, entities (and their attributes) and variables. These model components will be described as they are encountered in the model description below. The blocks are arranged in a graphical user interface and linked together to represent actual sequences of processes, logic-based decisions, etc. The end result bears a strong resemblance to a flow-chart. Additionally, Arena allows the model to be animated while running, giving users a chance to observe the flow of entities through the system and, if desired, live-updating statistics related to system performance. For more details on Arena’s capabilities, see their product website at <http://www.arenasimulation.com>.

II. Descartes Modeling Framework Overview

In late 2007, SpaceWorks Engineering conceived the idea to develop DES-based models for analyzing turnaround time, operations costs, DDT&E costs, and TFU costs. After being granted funding through a NASA Research Announcement (NRA) (see Acknowledgments for details), SEI began work in early 2008. In addition to the selection of Arena Basic Edition as the central DES engine, several other modeling conventions were laid out. Collectively, these decisions defined what was named the Descartes modeling framework. Over the three years of funding, the SEI team would use the Descartes framework in two phases. Phase I consisted of completely defining the framework structure and building a model for estimating turnaround time and recurring costs. Phase II focuses on a second model for estimating for DDT&E and TFU costs.

The structure of a Descartes model involves an Arena (.doe) model file along with several Visual Basic-enabled MS Excel 2007 (.xlsm) files. One Excel file has a user input worksheet, which is filled out with the vehicle design description and several modeling parameters. This file contains another worksheet in which the user inputs are converted into a number of variable arrays, arranged for input into the Arena model. Supporting this conversion is either a second .xlsx file or a set of worksheets in the same file containing a historical database of relevant data, collected to suit the needs of that particular model. Excel's native Visual Basic .NET (VB.NET) then opens the .doe file, and Arena's native Visual Basic for Applications (VBA) imports the values from the worksheet into Arena variable arrays. Both forms of Visual Basic will be referred to by the generic "VB" for the remainder of this paper. The model runs for a given number of replications (to increase the statistical validity of the outputs). After each replication, data is exported to a new worksheet in the output .xlsx file. After all runs are completed, this .xlsx file calculates averages and confidence intervals of the various metrics. This general file structure and data flow-path is followed for all Descartes models, allowing for consistent and reliable outputs that can be jointly utilized for more complete program analysis. To this end, it is a long-term goal for each Descartes model to be integrated into Phoenix Integration's ModelCenter framework, which allows the inputs and outputs from various design and analysis tools to be fed into each other. This simplifies not only initial analysis efforts, but also enables parametric sweeps and trade studies to be performed more efficiently. More specifically, we intend to make both primary Descartes models part of SEI's Integrated Risk and Cost Model (I-RaCM), a suite of cost and risk analysis tools integrated through ModelCenter.¹

A. Descartes-Hyperport

The first full-scale Descartes model is called Descartes-Hyperport, and is used for operations analysis. Hyperport requires a user to input approximately 100 vehicle design parameters, and about 15 program characteristics such as fleet sizes, available facilities, and frequency of long-term maintenance (or 'depot') visits. They then enter a labor configuration, with quantities assigned for various kinds of technicians (TPS specialists, propulsion maintenance specialists, etc). The model runs for a user-specified number of flights, incorporating learning curves, depot visits, and other forms of randomness to estimate turnaround time and ongoing cost. The simulation is repeated a large number of times so that statistically-valid averages and confidence intervals are the final output. The specific output metrics include estimated maintenance man-hours per flight, broken down by contributing subsystem; estimated processing times, again broken down by subsystem; and estimates costs per flight and costs per day, with breakdowns of several cost contributors, primarily labor, fuel, and spare parts.

Since the initial completion of version 1.0 in January 2009, Hyperport has been the centerpiece of SEI's operations modeling capabilities. Version 1.2, released July 2009, has already been utilized for several Air Force and NASA-sponsored vehicle concepts. In most cases, it was a part of a 'Level 1' operations analysis, comprised of Hyperport results for a stated desired flight rate, depot cost estimates, and a package of data including custom metrics highlighting primary operations cost drivers. This was integrated with reliability work and non-recurring cost estimates to give full life-cycle cost analysis to the customers sponsoring those efforts. Additionally, SEI has performed a 'Level 2' analysis of one of these vehicles, and supported a technology prioritization trade study on another. The Level 2 work will incorporate a wider exploration of trades using the Hyperport outputs as the key metrics, along with a higher level of fidelity in many of the supplemental estimates. Additionally, larger numbers of replications at Level 2 allowed further output of more complete distributions and/or percentile data, beyond the standard mean and confidence interval. For complete details of the Hyperport model, see reference 2.

III. Descartes-Origins

Once Hyperport was completed, the NRA-funded research effort progressed into Phase II, the modeling of DDT&E and production activities. The SEI team first had to set an appropriate scope for the new model. It was noted that a relatively large number of trusted, high-quality models exist for estimating up-front costs. Given the time and budget available for the study, it did not seem prudent to expend resources trying to replicate those capabilities in the Descartes framework. For this reason, the decision was made to take initial deterministic cost estimates as inputs, and have the new model take advantage of DES' probabilistic nature to distribute those costs over a predicted schedule, and to generate distributions of those costs due to schedule-related variability. This new model was given the name Descartes-Origins.

Origins development is not scheduled for completion until the end of 2010. Accordingly, this paper cannot be the final authoritative word on the model. However, from this point on, we will describe the model as it currently exists along with projections for how the remaining elements will be completed.

A. Origins version 0.7

Origins is built to model the various phases of design, development, and production of reusable launch vehicles. The first round of research focused on generalizing the design and development sequences used by NASA and the US Air Force (or, more generally, the Department of Defense). As shown in Fig. 1, the two agencies have varying checkpoints and design reviews, but the flow is easily generalized into distinct phases. Each of these phases has become a submodel in Origins.doe, with entities representing a concept design (later becoming the concept in production, testing, etc.) moving through them in sequence. Research was performed on a number of past programs to gain insights into how the design and production flow have evolved, and how closely they are adhered to depending on a variety of circumstances. The cases studied include successful programs with numerous launches, X-vehicles with very limited production, some programs that were cancelled entirely, and recent private efforts such as the Space-X Falcon 9. This wide variety of reference cases allows SEI to build a model that can be responsive to high-level decisions concerning funding levels, intended vehicle applications, levels of contractor involvement, sponsoring agencies, and whether or not the vehicles are to be manned. In response, the finished DES capabilities will be able to output probabilities of success, along with estimated cost and time distributions for both successful and unsuccessful programs.

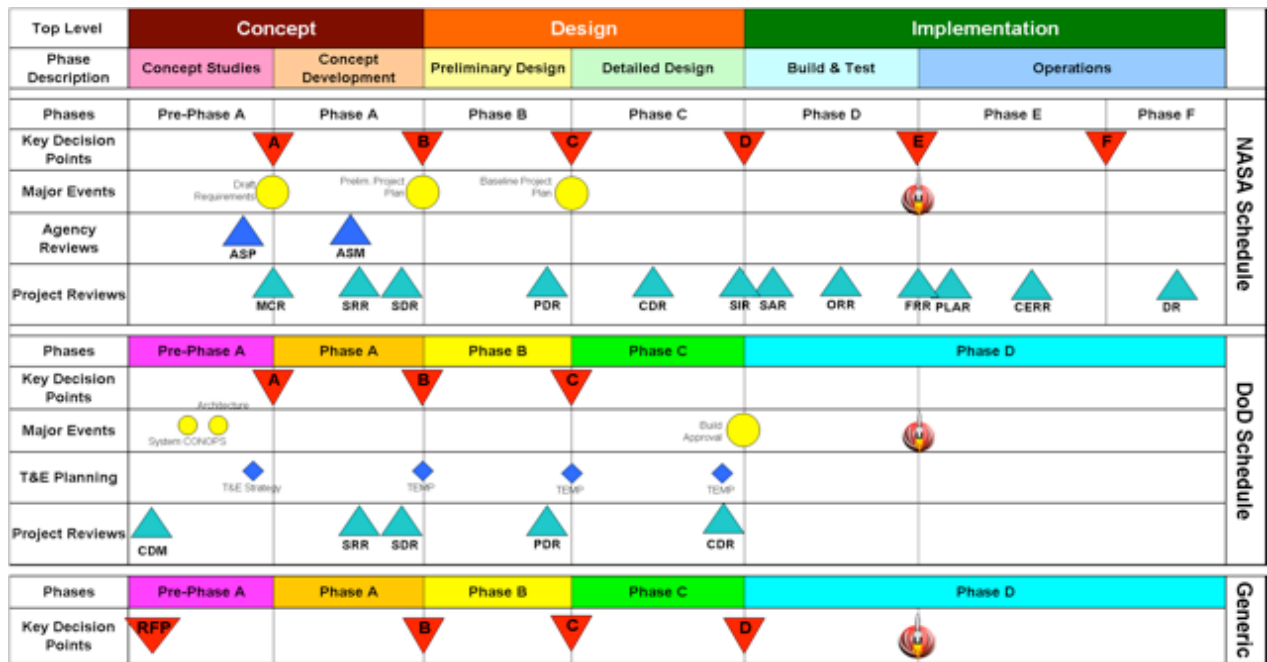


Figure 1. NASA and Department of Defense vehicle design processes

B. Origins Arena Model

Currently, the Origins model is split between two files (development phases and production are separate), but we will discuss them as if combined since that will be the final configuration. The top-level model, following the general structure shown in Fig. 1, consists of 5 submodels: Concept Studies, Concept Development, Preliminary Design, Detailed Design, and Production. Within each of these submodels are the Arena blocks representing the processes and decisions involved in that phase. The blocks within the four development phases (the ones other than production) all follow somewhat similar patterns. Fig. 2 shows the general phase layout, using Concept Development as an example.

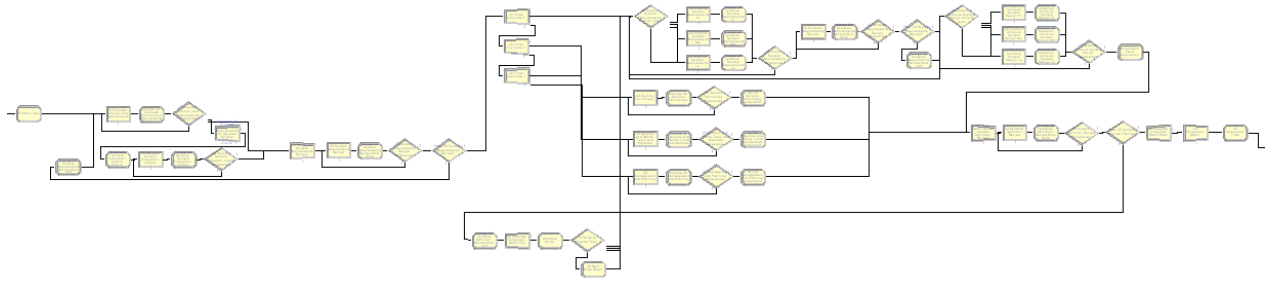


Figure 2: Screenshot of Concept Development submodel

Continuing with this example, an entity representing the vehicle concept enters from the left. An ‘assign’ block is used to record some information about this arrival into a variable array being used for data collection (more on this later). Then the entity is duplicated, with various copies going to processing flows that represent the various design disciplines involved at this phase. In the case of concept development, this list includes such items as geometry and packaging, structures and materials, sizing and closure, trajectory, propulsion, aerodynamics, aerothermal, subsystems, safety and reliability, maintainability and operations, cost estimation, and life cycle economics.

Each of these processing flows contains a set of various Arena blocks. A typical grouping is shown in Fig. 3. There are a total of 8 blocks in this group. The first, a ‘decide’ block, routes the concept entity to one of three ‘process’ blocks. The three process blocks represent three different levels of activity. For example, the ‘Hi’ block might imply 50 man-hours per day, the ‘Md’ block might imply 20 man-hours, and the ‘Lo’ block might imply 5. Each of these process blocks holds on to the entity for exactly one day. During this time, it ‘seizes’ the resources, in this case man-hours of funding, that it needs to get work done on the concept for that day. If the needed resources are unavailable, the entity will have to sit idle for a day or more until they become available.

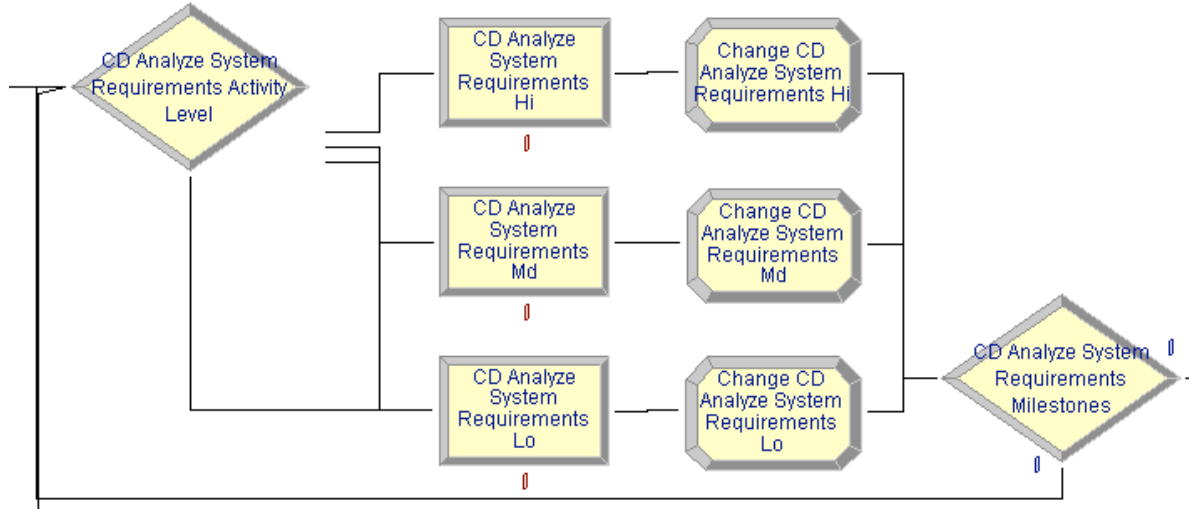


Figure 3: Typical set of blocks for a single development-phase process flow

At the end of the day, the entity passes on to an assign block, where several things are recorded. First, the quantity of resources used is recorded for cost tracking purposes. Second, the amount of work actually accomplished that day is calculated, in a two-step process. The quantity of resources used is multiplied by an efficiency coefficient, which is an input variable estimated from the characteristics of the team performing the work and the activity level they are working at. (In general, high and low activity levels, while sometimes necessary, are less efficient than the nominal ‘medium’ activity level for a number of reasons.) Second, based on a calculated input probability variable, there is a chance of a ‘random error’ occurring. The error, once generated, takes on a random size generated from a probabilistic distribution with the odds favoring ‘small’ errors. In the context of concept

development, a small error might be a computer crash that loses half a day's worth of work, while a larger area might occur if, for example, the aerothermal analysis reveals that the trajectory being will require a new kind of thermal protection material, which requires adjusting the weight statement and re-closing the vehicle. These errors, once generated, will increase the work remaining for this (and possibly other related) process flows.

After adjusting the 'work remaining' variable for the process, the entity moves to the second decision point in the set. This decision queries whether the 'work remaining' has reached 0. If so, the entity will move on from this point and enter the next flow on the next day. If not, it will cycle back to the first decision block and, the next morning, be routed to one of the activity levels again. It is also possible that certain intermediate milestones could trigger other routing options. For example, if operations assessment cannot take place until reliability assessment is about 50% complete, the second decide block in the reliability process would contain an additional possibility: if 'work remaining' is below 50% of the original quantity of work, a duplicate entity would be sent along to begin operations analysis while the original continued to go through reliability assessment.

1. Non-standard Development Processes

While the above-described flow is utilized for most analysis processes, there are certain processes for which alternative model structures are required. Some processes such as management and oversight may operate at near-constant activity levels, so they do not require the three pairs of process and assign. Additionally, review activities are modeled as a one-time process (rather than a daily repeater) and require a different kind of feedback loop. After a review, a concept may pass and move on to the next phase, it could pass with changes, or it could fail. If a concept does not pass outright, it will move to an assign block where work is re-added to various process areas within that phase and then the concept will pass back through all those processes until the 'work remaining' reaches 0 again. A 'pass with changes' condition would add some small percentage of work back on to a handful of processes, and after completion the concept would route directly to the next phase. A 'fail' would result in work being re-added to all processes and the concept being subjected to another round of review, although on the second attempt it is given a higher chance of success. If a concept fails review twice, it fails completely and the simulation run terminates.

2. Production Processes

The production submodel is structured very differently from the development phases. The 8-block process model works well to represent research and development-type tasks, where for the most part the resources are people's time, with limited flow of materials. Production, on the other hand, uses a more traditional DES approach. Process blocks are used to represent the fabrication (or acquisition) of various subsystems, and additional processes are used for the assembly of those subsystems into the vehicle. For a single stage of a vehicle, some or all of the following may be modeled: tanks, primary structures, landing/recovery systems, thermal protection, electronics, power generation and distribution, actuation, crew compartment, attitude control systems, and propulsion. Each of these subsystems might be produced 'in-house' by the primary contractor or they may be outsourced, but some random distribution applies in either case. Once subsystems are available, they are assembled in a logical sequence that may vary depending on the vehicle configuration.

For now, only a single instance of this production flow exists; it is shown in Fig. 4. By model completion, there will be partial duplicates of this flow to allow for vehicles with up to 3 stages, and to allow for production and testing of prototypes prior to operational flight unit production. Prototype production will likely be triggered by events within the detailed design block, rather than waiting for that phase's completion. Additionally, prototype testing could result in feedback to the detail design processes in case unexpected changes must be made in between prototype and operational unit production. Finally, it will be possible to produce multiple vehicles (and multiple prototypes, if desired), with the schedules and costs tracked independently.

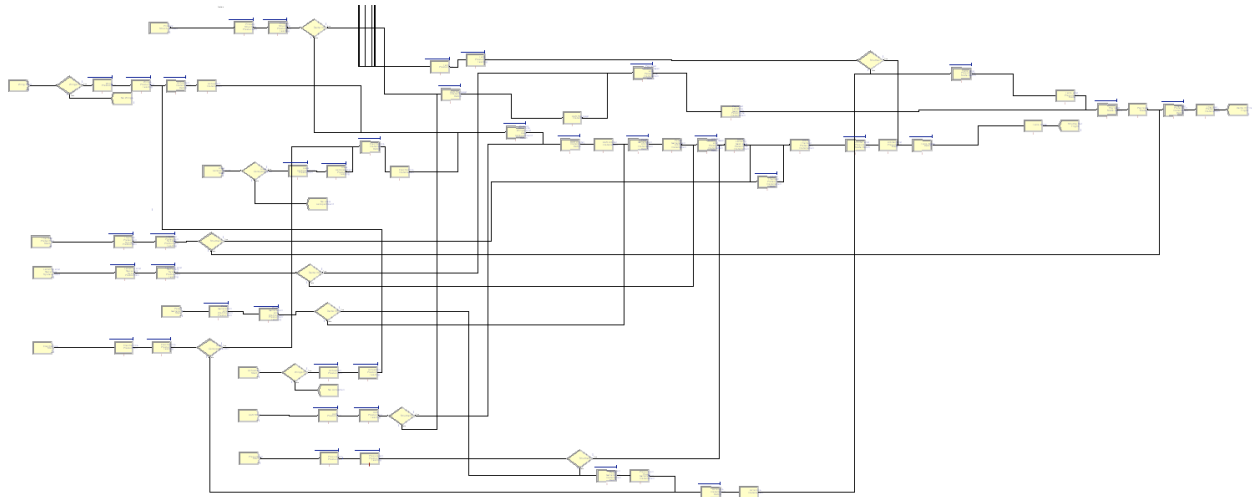


Figure 4: Screenshot of a single stage production process flow.

C. Origins Excel Input and Output Files

Throughout the description of the Arena model, several references were made to variables that were calculated and input into the simulation. Estimated process times, resource requirements, process efficiencies, review success probabilities and the like are all calculated before the Arena model begins to run. In an Excel input file, a user must answer a number of questions about the vehicle configuration, a number of questions about the various teams performing different aspects of the program, the deterministic cost estimates generated from an external model, estimates of the program budget, and several simulation parameters. Based on data collected over the last year, these five sets of inputs generate numerous tables of numbers describing all of the variables mentioned above. In most cases, the shapes of the distributions of these variables have already been hard-coded into the model, but the location remains to be specified. Visual Basic code within the Excel file performs all of these calculations then opens the Arena file and tells it to begin. The VB within the Arena file then calls on the worksheets of the Excel file to fill in those data tables within the DES model.

After each replication of the Arena model, all of the output data that has been collected (covering various things mentioned above but generally consisting of cost and schedule data) is exported into another Excel file. This file then uses a series of pivot tables to sort the data, and statistics are generated from it. The final outputs include breakdowns of costs and schedules, estimated program success chances, and some graphics highlighting the specific relationship between schedule slippages and cost overruns. Armed with this information, a user could suggest various changes to the vehicle design or program strategy (including budget) that might improve the ultimate result. Depending on the goals of a particular program, one could focus on improving median estimates, or on reducing uncertainty by trying to minimize some other percentile of the output.

Upon integration with ModelCenter these improvements could be optimized automatically and, combined with Hyperport runs, dynamically recommend vehicle improvements to the design team based on a complete picture of life cycle metrics.

IV. Conclusion

The Descartes project's goal is to be able to consistently estimate reusable launch vehicle project costs and timelines, using a single coherent modeling philosophy. The ability of discrete event simulation to model complex systems with numerous process flows and interaction of resources makes it an ideal choice for this type of work. Descartes-Hyperport has become a working tool that is being employed on an ongoing basis. Hyperport has performed well as a proof-of-concept, producing high-level estimates in line with expert opinions, while also giving insights into lower-level cost drivers and sensitivities that may not have previously been available. Descartes-Origins will be able to take advantage of lessons learned and modeling techniques perfected while developing Hyperport to become an equally helpful tool contributing to life-cycle cost analysis. Once completed in late 2010, Descartes-Origins will be able to give probabilistic estimates of DDT&E and TFU costs in a similar form to those currently available from other tools, but they will be coupled with time estimates and chances of success, giving a more complete picture than is currently available of the lifetime potential of a future program.

As the Space Shuttle fleet is retired and a variety of private companies and government programs work towards the next generation of reusable launch vehicles, decisions are likely to be made based not only on vehicle performance, but also on cost, reliability, and long-term maintainability projections. The tools developed at SEI during the Descartes program will be able to provide reliable, detailed estimates, giving programs the data they need to make those decisions. Hyperport is already supporting this objective, with Origins following soon.

Acknowledgments

Sponsorship and financial support for major portions of this project are provided by NASA's Aeronautics Research Mission Directorate under 2007 NASA Research Announcement contract NNL08AA30C. The authors would specifically like to thank the Contract Officer's Technical Representative Ms. Lakisha Crosby, the hypersonic technical point of contact Mr. Jeffrey Robinson, Mr. John Martin, and all of NASA Langley Research Center.

The authors gratefully acknowledge the contributions of colleagues at SpaceWorks Engineering, Inc. who contributed to this paper. Intern contributors included Mr. Brendan Dessanti, Mr. Luke Walker, and Ms. Aimee Sostilio, who all supported research for the Origins model, and Mr. Eammon Bemis, who assisted in the early Arena model development. Mr. William J.D. Escher (SEI Affiliate) and Mr. Carl Ehrlich (SEI Affiliate) provided sage advice as well as references relevant to operational aspects of past aerospace programs.

References

¹DePasquale, D., Charania, A., "I-RaCM: A Fully Integrated Risk and Life Cycle Cost Model," AIAA-2008 7703, Space 2008, San Diego, California, September 9 - September 11, 2008.

²Kelly, M., Charania, A., Olds, J., and DePasquale, D., "Discrete Event Simulation of Reusable Launch Vehicle Development, Acquisition, and Ground Operations," AIAA-2009 6484, Space 2009, Pasadena, California, September 14 - September 17, 2009.