

IAC-08-D2.2.9

ACTIVITY-BASED SIMULATION OF HYPERSONIC REUSABLE LAUNCH VEHICLE TURNAROUND TIME

A.C. Charania*

SpaceWorks Commercial, a division of SpaceWorks Engineering, Inc. (SEI), United States of America (USA)
ac@sei.aero

Dominic J. DePasquale^t

SpaceWorks Engineering, a division of SpaceWorks Engineering, Inc. (SEI), United States of America (USA)
dominic.depasquale@sei.aero

John R. Olds^t

SpaceWorks Engineering, a division of SpaceWorks Engineering, Inc. (SEI), United States of America (USA)
john.olds@sei.aero

ABSTRACT

Future reusable launch vehicles are considered to have potential benefits compared to expendable systems. These benefits can arise from reductions in turnaround time, recurring operations costs (labor and materials), and life cycle cost. Yet quantifying these operability and affordability effects, in some manner beyond using historical analogies, has proven difficult for the systems analysis community. SpaceWorks Engineering Inc. (SEI) has developed, through research and quantitative modeling, an activity-based simulation to predict various operational metrics for future hypersonic space access and point-to-point reusable launch vehicles. This model is implemented in the discrete-event simulation software Arena, commercially available from Rockwell Automation. A database of various activities associated with turnaround operations has been developed. Inputs for the analysis model include descriptions of the basic vehicle concept of operations, mission models, and performance information. The current model outputs turnaround time. Future versions of the model will output recurring operations cost as well development and acquisition cost. 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 to model reusable launch vehicle operability and affordability.

NOMENCLATURE

ABS	Activity Breakdown Statement	KSC	Kennedy Space Center
DES	Discrete Event Simulation	LCC	Life Cycle Cost
DDT&E	Design, Development, Testing & Evaluation	LEO	Low Earth Orbit
GSE	Ground Support Equipment	LOC	Loss of Crew
		LOM	Loss of Mission
		NAFCOM	NASA Air Force Cost Model
		NASA	National Aeronautics and Space Administration
		OPF	Orbiter Processing Facility
		RLV	Reusable Launch Vehicle
		SEI	SpaceWorks Engineering, Inc.
		SOA	State-of-the-Art
		STS	Space Transportation System
		TAT	Turnaround Time

* - *President, Senior Member AIAA.*

¹ - *System Engineer, Member AIAA.*

^t - *Principal Engineer, Associate Fellow AIAA.*

TFU	Theoretical First Unit [Cost]
TPS	Thermal Protection System
USAF	United States Air Force
VBA	Visual Basic for Applications

INTRODUCTION

Overview

In order to achieve the breakthroughs of more responsive and cheaper global reach and space access, launch vehicles have to be reusable. To obtain the most benefit from such reusability, space launch assets should be easy to turnaround from one launch to another, as well as inexpensive to operate. Operational benefits are often cited for hypersonic air-breathing propulsion versus rocket or solid propulsion systems. These benefits arise from reductions in the following: turnaround time (for reusable systems), recurring operations costs (labor and materials), and total life cycle cost. Yet quantifying these operability and affordability effects for these reusable launch vehicles (RLVs), in some manner beyond using historical analogies, has proven difficult for the systems analysis community. The current generation of analysis tools must go beyond the bounds of traditional historical-based relationships and calculate outputs on actual activities needed for particular subsystems.

The Space Shuttle Example: Expectations versus Reality

The United States Space Shuttle was initially conceived of in the 1970s as a reusable successor to America's fleet of expendable launch vehicles. The Space Transportation System (STS), as it is known, was supposed to be a more cost effective way of reaching Low Earth Orbit (LEO). There were visions that the Space Shuttle would be a system that even with some expendable elements (external tank and solid rocket motors), would be relatively easy to turnaround from one flight to another, with minimal maintenance, inspections, or servicing. It was envisioned that minimal Ground Support Equipment (GSE) would be required, payload interfaces would be simple, and that overall flight rates would be very high (more than ten flights per year).¹

In 1976 estimates included the following: "The Space Shuttle Orbiter is designed for a 2-week ground turnaround, from landing to relaunch. About 160

hours of actual work will be required."² The reality of the STS turnaround experience has not been so optimistic (see Fig. 1). For instance, the average turnaround time for a shuttle orbiter in the Orbiter Processing Facility (OPF) from 1990 to 1997 was 88 days.³ It takes several months to inspect, turnaround, and integrate a complete space shuttle stack. In addition, the orbiter may be required to undergo depot level maintenance activities that may take it offline for much longer periods of time.

Advanced Simulation Modeling for Space Transportation

There is a need for a simulation-based model focused on specific hypersonic systems (such as space access vehicles or high speed point-to-point missions) that can analyze subsystem (e.g propulsion, thermal protection) operational activities to generate outputs related to time and cost. Inputs for such an analysis model include descriptions of the basic vehicle concept of operations, mission models, and performance information (including overall system weight assessments).

There have been several attempts at simulation-based modeling and, more particularly, in using discrete-event simulation (DES) to model such space systems.^{4,5} These include a few projects in various stages of development by NASA.^{6,7,8,9} However some of these efforts have limitations that make them not directly applicable to hypersonic systems analysis. Some of these limitations include:

- 1) Focused on traditional rocket elements such as Space Shuttle or NASA Ares I/V transportation architectures
- 2) Limited to NASA Kennedy Spaceport facilities and operations
- 3) Potentially too broadly address overall spaceport operations and neglect the nuances of specific subsystem turnaround activities like hypersonic propulsion integration

Traditional systems analysis tools and models have utilized historical data to predict metrics for air and space systems. However, these philosophies of modeling have difficulty when attempting to model new systems with new paradigms of development and operations. For instance, NASA Kennedy Space Center (KSC) has attempted several simulation model activities focused on KSC exclusive facilities

and NASA specific exploration related launch systems (Space Shuttle, Ares I/V).

There is a need to focus on the areas of operability and affordability for hypersonic vehicles (and particularly their propulsion subsystems). Many of the existing NASA-led activities focus on airframe or spaceport operations and may tend to relegate propulsion activities to a lower priority. There is a need to do a better job in modeling specific propulsion operational activities (especially for hypersonic launch systems) using the actual activities associated with those technologies. This will help to better quantify metrics for turnaround time, recurring operations cost, and overall life cycle cost.

In addition, even though there has been activity on developing simulation models of turnaround operations, there has been limited development of simulation-based models of development and acquisition cost, a slightly harder problem. Government models such as the NASA Air-Force Cost Model (NAFCOM) or commercial models such as Price/SEER are currently employed by cost analysts to generate Design, Development, Testing, Evaluation (DDT&E) and Theoretical First Unit (TFU) costs for aerospace systems (including launch vehicle systems). For many of these models, the approach to accounting for program unique methods of development and production is to apply multiplicative complexity factors to the output cost estimates. A better method is needed to accurately model any innovative or alternative method of production in order to better account for the supply chain and its impact on DDT&E and TFU costs (and ultimately life cycle costs).

In general, better methods need to be developed to model the following metrics: DDT&E cost, TFU cost, recurring operations cost, and turnaround time. These new methods need to be able to simulate the actually envisioned development and operations, not just make corrections to analogies of past historical programs. Determination of such output metrics could be improved if actual characteristics of the system and manner of development were accounted for in an activity-based costing methodology. 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 operability and affordability.

DISCRETE EVENT SIMULATION (DES)

Overview

Discrete Event Simulation (DES) is a kind of simulation sometimes also referred to as the event scheduling approach. In DES, complex systems and processes are represented by the chronological sequence of events that comprise them. These events may be arranged in series or in parallel when modeling a system. Each event is defined as occurring at a single point in time and effects some change on the system. Any non-instantaneous events are defined by their start and end times, which would be listed as two separate events. A program running a simulation keeps a list of future events that have been scheduled, and adds to that list as new events get scheduled. For example, when a 'begin process' event takes place, the length of the process is determined and the 'end process' event is added to the schedule. Once all changes associated with an event take place, it is removed and the simulation moves on to the next scheduled event.

This approach to simulation offers several advantages. First and foremost, it is very computationally efficient. The processor does not have to monitor various ongoing activities at once, only to handle a single event at a well-defined moment. It is also surprisingly flexible, since just about any kind of process can be broken down into a series of points at which conditions experience a discrete change. Even an assembly line can be defined by when each item passes a series of relevant points, such as when it is handed off from one technician to the next. The actions taking place at each event can also be as complex as needed to properly express what changes are taking place, and multiple new events could be scheduled at once. Events can involve random distributions if needed, and both inputs and outputs to an event's calculations can be based on any number of conditions whether specific to an object being processed or background conditions.

DES is commonly applied to model operations processes and has been implemented by a wide variety of companies and other organizations in the automotive, manufacturing, aerospace, construction, energy, banking, shipping, health care, and other industries. Just some of the organizations using DES are Pepsi Americas, Ford, Bayer Corporation, UPS, Dow Chemical, the U.S. Army, and the U.S. Air Force.¹⁰ Examples of aerospace specific applications

include NASA Kennedy Space Center for spaceport operations simulation, Delta Airlines for optimization of Atlanta airport check-in queues, Lufthansa for scheduling airport hub operations, and Lockheed Martin during design of the Joint Strike Fighter.^{11,12,13,14}

Arena Tool Description and Capabilities

Simulations can be performed at a variety of levels of modeling complexity. Originally, computer simulations were written in general-purpose languages. As spreadsheets including Microsoft Excel became more advanced, they became useful for modeling some kinds of Monte Carlo simulations. Special-purpose programming languages including GPSS, SLAM, and SIMAN were eventually developed specifically for simulation. Rockwell Automation took the SIMAN language and built Arena on top of and around it. Arena is a graphical user interface but with access to numerous levels of modeling.¹⁵ A user can create animations and interfaces with other programs (including Excel), all while modeling a wide variety of systems and processes

The simplest way to use Arena is to use the graphical interface and build models from process 'blocks,' which are objects representing various parts of a system and are arranged into patterns resembling standard flow charts. These 'blocks' are built out of SIMAN code which is generally kept hidden from the user. During the simulation run, 'entities,' representing physical objects that are processed in some way, pass from one block to another. These entities have 'attributes,' potentially including such things as arrival time, serial position in a sequence of entities, or any physical characteristics of the entity. 'Resources' could be facilities, labor, equipment, or anything else that is occupied by an entity at some point during the process. 'Variables' are other numeric (or text) values that are maintained in the background but can be changed or read by entities during some kinds of processes.

The Arena Basic Edition allows the user access to the 'Basic Process panel,' a collection of available blocks as well as some spreadsheets to manage things like variables that can be defined externally to individual entities or processes. The basic process blocks are:

- 1) Create. This is the origin of any entities. They can arrive randomly or according to a

- deterministic schedule, and in batches or as individuals.
- 2) Dispose. This is the opposite of a create block, and all entities that have completed their purpose are sent to a dispose block.
- 3) Process. This is the where any time and resource-consuming activities take place. These blocks can be written to take deterministic or random amounts of time, and to base times on entity attributes if desired.
- 4) Decide. Decisions can be based on random chance, entity attributes, or the current value of a variable. The block sends an entity in different directions depending on the outcome.
- 5) Batch. Entities can be combined into a batch, after which they are treated as a single entity.
- 6) Separate. Batches can be split back into their components, or single entities can be duplicated to allow for parallel processing.
- 7) Assign. Assign blocks can change the values of attributes or variables.
- 8) Record. Record blocks can track time intervals, count passing entities, or record values of attributes or variables when an entity passes through them.

Additionally, a user can insert a 'submodel' along with these blocks. Appearing to act in some ways like a Process block, a submodel is essentially a folder in which any variety of blocks are placed, generally as a modeling convenience.

Each of the above blocks is activated when an entity arrives (except Create), after which it carries out whatever instructions it has, then passes the entity along. Create, dispose, decide, separate and record are all always instantaneous. Batching can take time while one entity waits for others to arrive with which it can be batched. Process blocks can take time and also occupy resources for the time they are active. All of these blocks have additional functionality not described here but available in reference documentation.¹⁶

While all blocks used in SpaceWorks Engineering's models are from the list above, additional panels available in the Arena Professional Edition allow a wider variety of material handling, file input/output, and process coordination functionality, as well as allowing insertion of individual SIMAN blocks into the model. The equalizing factor in all Arena licenses

is access to a Microsoft Visual Basic for Applications (VBA) editor. Code can be written in VBA to perform a wide variety of functions both within an Arena model .doe file and within other programs that are VBA-compatible. SpaceWorks Engineering uses VBA extensively to coordinate multiple files, user input/output, and generally control the flow of a model.

DESCARTES – HYPERPORT

Overview and Development Plan

SpaceWorks Engineering conceived of the Descartes DES framework for operations and cost analysis of multiple aerospace design concepts, ranging from transportation to infrastructure. Descartes will be built upon the commercially available DES software product Arena, by Rockwell Automation. Descartes consists of the simulation model, databases of input and output interfaces, and custom programming in VBA to augment Arena built-in analysis capabilities. When Descartes is applied to model a particular aerospace system, a customized instance of Descartes is created and that instance is named according to the system of interest. The instance of Descartes used for activity-based simulation of future hypersonic reusable launch vehicles, the subject of this paper, is called Descartes-Hyperport.

Complete development of Descartes-Hyperport is a three year, two phase project. The objective of the project is to develop, through research and quantitative modeling, an activity-based simulation that can predict turnaround time, recurring ground operations cost, development, and acquisition cost for future hypersonic flight vehicles spanning the range of missiles, space access vehicles, and hypersonic cruise vehicles. This development will go beyond traditional historical modeling efforts and focus on modeling the actual activities associated with the propulsion system. The final result of this framework will be to help inform decisions makers of the impact of their technical (i.e. TPS type, propulsion system, etc.) and programmatic decisions on operability and affordability metrics (turnaround time and cost).

Fig. 2 illustrates the various components of the project. Phase I will consist of an initial development of the turnaround analysis and recurring operations cost analysis capability of the simulation. Phase II will consist of creating activity models to determine Design, Development, Testing, and Evaluation

(DDT&E) and Theoretical First Unit (TFU) costs. Each phase begins with extensive research on historical and current State-of-the-Art (SOA) modeling philosophies, focused as appropriate on operations turnaround time, recurring operations cost data, and development/acquisition cost. Towards that end, the team has initially developed an Activity Breakdown Statement (ABS) in MS Excel consisting of over 1000 activities identified in 12 activity categories (Cargo Processing, Traffic Control, Launch, Landing, etc.) developed from NASA KSC organizational categories.¹⁷ The ABS was then refined through additional research on other aerospace operational processes. During each phase, the team will apply its models to case studies. These case studies will be specifically focused on near term and far term hypersonic reusable launch vehicles (RLVs) consisting of multi-stage to orbit space access vehicles.

Descartes-Hyperport is designed so that it can be integrated into the ModelCenter® multi-disciplinary analysis framework. ModelCenter® is developed and commercially marketed by Phoenix Integration, and is available as an executable for the Microsoft Windows® operating system. The software allows for the coupling of various disciplinary software tools into one integrated model. It provides a user-friendly graphical front-end and various built-in capabilities for linking the tools together (both feed-forward and feed-back links), performing trade studies, and collecting and analyzing data from a large number of executions of the integrated software tools. ModelCenter® integration greatly increases the capacity for operations analysis using Descartes-Hyperport.

SpaceWorks Engineering has extensive experience developing multi-disciplinary models for analyzing space systems applications such as reusable launch vehicles, lunar exploration systems, and human Mars missions. These integrated models have included traditional performance disciplinary analysis tools (e.g. propulsion, aerodynamics, mass estimating), economics, reliability, and operations tools. Descartes-Hyperport and other implementations of Descartes will enhance operations analysis capability for future integrated models. Descartes-Hyperport will also become part of SpaceWorks Engineering's Integrated Risk and Cost Model (I-RaCM), a suite of newly developed and existing industry-standard cost and risk analysis tools combined within ModelCenter® allowing for rapid evaluation of life cycle cost, operations, reliability, technology

development costs, and commercial business case viability.¹⁸

In order to integrate Descartes-Hyperport into ModelCenter®, the interface between the two tools must be defined. The interface between ModelCenter® and each software tool is defined by a wrapper file. The wrapper file contains code instructing ModelCenter® how to transfer input/outputs and execute the integrated software tool. Development of a wrapper file and testing of Descartes-Hyperport within ModelCenter® is part of the Phase II activities.

Descartes-Hyperport Functional Layout

As mentioned above, the central element of a Descartes model is an Arena (.doe) file. This file contains a virtual representation of the operations being modeled, and when it is executed it simulates numerous replications of those operations. While in general Arena models can be manipulated directly by the user, one of the strengths of Descartes lies in a simplified user interface based in Microsoft Excel. To fully understand the workings of Descartes one must understand both how the Arena model works on its own as well as how it is manipulated through the Excel files.

User / File Interactions

Descartes-Hyperport is a single model built with an Arena file and a pair of MS Excel workbook files. The Arena model, Hyperport[version].doe uses Visual Basic for Applications (VBA) to interact with the Excel files. One file, Hyperport.xls, acts as the user interface in which various aspects of the vehicle in question are selected, and the end results are displayed. The other, DescartesData.xls, is a database of historical maintenance and operations requirements for various subsystems from a variety of relevant vehicles. Fig. 3 is a schematic of these relationships.

The first file opened by a user is Hyperport.xls. In the first worksheet of this file, one can input various design parameters for the vehicle and vehicle subsystems. Subsystem-level design parameters vary by subsystem, but in general define aspects of the design relevant to operations activities. For example, inputs for the Thermal Protection System (TPS) subsystem include selection of materials and entry of component surface area. In addition to design parameters, various resource inputs are required such

as labor availability at the turnaround facility. Finally, simulation inputs such as the length of individual simulation runs, the number of replicated runs, and the vehicle arrival characteristics must be selected.

After fully specifying the vehicle to be studied and the other parameters mentioned, the user closes Excel and opens the Arena model file. Clicking ‘run’ will activate the VBA code in Arena which does several things. First, the inputs from Hyperport.xls are imported. Second, these values are used in reference to the second Excel file, DescartesData.xls. DescartesData acts as a database, with separate worksheets for various standard subsystems. Each sheet has historical data about operations and maintenance of that subsystem, from various reference vehicles and with multiple parameter inputs possible. The VBA code looks up the relevant subsystem data and uses it to populate the times and resource use levels for the process blocks in the Arena model.

The next VBA action is to re-open Hyperport.xls and add a new output sheet to the workbook. Arena then runs the user-specified number of replications and, upon collecting statistics from that replication, writes them into a column on the output sheet. These statistics include breakdowns of times and costs from the different parts of the operation. After the simulation terminates, summary statistics are calculated in Excel from the individual replication data, giving averages and the half-widths for 95% confidence intervals around those averages. The sheet has a timestamp and also includes a copy of the input sheet (found in the rows below the output). This allows a user who wants to change some input parameters and run the simulation again to go to the input sheet, make the changes, save the file, close it, and run Arena as before. The new run creates a fresh output sheet, so comparisons can easily be made between runs with different inputs.

The layout of the Arena model shown in Fig. 4 is meant to resemble the layout for a hypersonic maintenance facility, a ‘hyperport.’ The orange icons represent the various activity areas. The number adjacent to each icon tracks the number of vehicles currently in process at that area. These areas may correspond to different buildings, and the overhaul depots may even be located at a separate site, so the spatial relationship is unimportant but helps the user visualize the flow of activity. The Create block where vehicles start the analysis process is actually hidden

behind the runway, and from there the flow leads to the 'Landing/Safing' area. A vehicle proceeds through the various areas as needed, and eventually goes through 'Preflight' before going back to the runway and exiting the model through the Dispose block (also hidden.)

With the exception of the Create and Dispose blocks, the rest of the functionality of the model is built into a structure of submodels. Each submodel represents a major facility or activity area of Descartes-Hyperport (e.g. Lower Stage Turnaround Facility, Lower Stage Depot, Integration Facility). The logical structure of a submodel is accessed by double-clicking a small block directly above the relevant icon. Within the submodels the vehicle (or vehicle stage) enters from the left, passes through Assign blocks noting area arrival time and any other relevant data, then passes into the blocks representing the various maintenance processes themselves. The last few blocks in each submodel record statistics for that activity area, then account for transportation time to the next area. A duplicate 'dummy' entity is also created right before the transport process block. This dummy entity occupies the facility and labor resources long enough for the facility to be cleaned, re-stocked, and prepared for the arrival of the next stage, then it is disposed.

When the model is run, if animation is turned on, the user can view icons representing vehicles and stages 'landing' at the runway and moving from area to area. If the user clicks to open any one of the submodels, he/she can similarly observe the flow within the submodel between individual blocks. Because of the nature of discrete event simulation and the Arena software, the speed of the animation has no bearing on the simulation itself, and is in essence purely aesthetic. In all cases, any actual processes are happening instantaneously and the simulation is paused during animation. Current simulation time can be seen in the lower-left corner of the Arena window.

While it is running, the view of the whole model also includes some statistical displays. In addition to the current number of vehicles by facility, the graph to the left tracks the running total of vehicles in the whole system. The red number below it shows the average, of those vehicles completed so far, of the total turnaround process time. All of these displays reset between replications. After all replications are complete the simulation will stop and all statistics will be visible in the new spreadsheet of the Excel

file. Some statistics can also be viewed in standard Arena reports, but the output Excel file post-processes, filters, and organizes the most important outputs in a way that is most convenient for the user.

Current Model Status

As of the writing of this paper, Descartes-Hyperport is in a state of ongoing development. All features described in the Model Structure section exist in an interim form, and are being improved by the development team at SpaceWorks Engineering. Completion of this turnaround model is scheduled for December 2008. The general structure of all the files is already in place. The main work is currently directed towards researching and developing each activity area and the processes within them, modeling them based on historical processes, and adding relevant data points to the DescartesData workbook.

The Arena model Hyperport.doe currently contains the following submodels: Landing/Safing, Demate, Turnaround for each stage, Airframe Depots and Propulsion Depots for each stage, Integration, and Preflight. Each submodel contains a set of 'process' and other blocks meant to represent the processes needed to perform maintenance on a hypersonic vehicle. Some decisions have been made to accommodate modeling of a two stage vehicle concept, but the general effort has been toward making the model as flexible as possible for a variety of hypersonic vehicle configurations.

At the first area, Landing/Safing, there are blocks accounting for landing and taxi time, followed by vehicle safing, passenger egress and any payload removal. There are separate sequences of blocks depending on whether the vehicle arriving is a lower stage, an upper stage, or the two combined after an abort. An upper stage always lands within a few days of a corresponding lower stage, with exact parameters set by the user, and mission aborts occur randomly, at an average rate set by the user.

If the stages do land together, they leave Landing/Safing and go to the Demate area, where the model accounts for time and resources involved in separating the two, then sending them back to the usual path leading to the Turnaround areas. Each stage has its own turnaround area since their maintenance needs are likely different. The two are modeled very similarly, with primary differences being the parameters on the individual processes within. In each case, the more complex and

time/resource intensive processes have been given their own submodels to allow more in-depth modeling while keeping the interface a manageable size. Turnaround activities are commonly grouped into submodels by subsystem. The TPS subsystem of each vehicle stage is one example of such a submodel. Based on user-input data on the types and sizes of the various TPS components on the stage, several ‘copies’ of the stage are processed in parallel, with times based on historical TPS maintenance data. Other second-level submodels are being developed with similar structures, including processes in serial or parallel as determined by historical maintenance data.

The logical structure of a Turnaround facility submodel is shown in Fig. 5. The first process in the Turnaround facility is a post-flight inspection of the stage. If the airframe or propulsion system fail the inspection, currently determined randomly, they are sent to the respective depot for more long-term overhaul-level maintenance. The structure of these submodels is similar to the turnaround models but with less in-depth modeling. It is possible to make these depot visits part of a regular schedule as well as occurring after random failures. After completing the overhaul, the airframe or propulsion system is returned to the turnaround area where it is combined with the other system and re-inspected before being sent to stage integration.

The function of the Integration area is generally analogous to the Vehicle Assembly Building in the STS program. For the current concept vehicle, it is assumed the two stages will be kept in a horizontal position. The submodel includes processes performed on each stage as it arrives at this area to prepare for integration, the integration itself, and the loading of any payload to the combined vehicle.

The final submodel is the Preflight area. Here the vehicle is fueled, the crew board, and any other activities taking place in the last few hours before flight take place. This submodel also includes process blocks for the taxiway and takeoff of the vehicle and collection of all final statistics. The simulated disposal of the vehicle entity takes place just after this area. As with many parts of the Arena model, at this time there are many individual process blocks in place and more are likely to be added as the development continues and data becomes available.

Data Plan

The key to the long-term usefulness of Descartes in general, and Descartes-Hyperport more specifically, is the population of the spreadsheets in the DescartesData Excel file. As many previous efforts have been made by various teams from NASA, the USAF, and others to model the turnaround and maintenance operations for both high-speed atmospheric and space-bound vehicles, the goal of the Descartes team has been to gather the best available data from these previous studies and combine it for use in this model. In every case possible, the data from one or more sources will be put into the DescartesData sheets, which are called by the Arena model to determine times and resources for many of the Process blocks in the model.

At this time, the team is in the process of determining which sources contain the forms of data most useful for inclusion. Most time series data explored so far has consisted of single point averages for various processes. Since DES is a probabilistic process, this requires the team to make certain assumptions about probability distributions surrounding these averages to fill in the model. Generally individual process blocks are modeled with triangular distributions. While real times might be better fit to a normal distribution, triangular distributions eliminate the potential problem of negative values generated at the left end of normal curves. Resource use data is simpler since it tends to be more deterministic, but consideration must be given to factors including maximum feasible crew sizes due to physical limitations of the workspace, as well as labor time utilization and potentially variable labor schedules. Current potential data sources include:

- 1) USAF’s SOVOCS-SI, a collection of spreadsheets referenced in a ModelCenter model
- 2) NASA Kennedy Space Center’s LLEGO, an Excel-based analyzer originating from the shuttle program
- 3) NASA Langley Research Center’s RMAT, with useful data on engines
- 4) NASA Ames Research Center’s TPS-X, containing extensive data on numerous TPS materials
- 5) NASA Marshall Space Center’s Comet / OCM
- 6) NASA Kennedy Space Center’s Root Cause Analysis database, containing very detailed

time and resource data from a single STS mission turnaround

- 7) Data from the X-15, X-34, and SR-71 programs
- 8) Various AIAA papers and NASA technical reports^{19,20,21,22}

Whenever possible, multiple data sources will contribute to a single subsystem spreadsheet in DescartesData. These multiple sources will allow either a better-fit probability distribution, or potentially the ability for a user to specify more detail about their vehicle and choose which data source is appropriate based on the design.

FUTURE WORK

Future work on the Descartes tool will include continued development of the Arena-based turnaround time and recurring operations cost models for Phase I of this project. The team will continue definition of resources (facilities, GSE, personnel types) within the Arena model for turnaround operations for multi-stage space access vehicles. Additional work will continue on research of various process times and costs for use in the model. This will help in development of subsystem time inputs based upon qualities of the vehicle concept. General research will continue on operational processes for previous concepts. Additional research will also be performed to update the refine mapping of activities to the Activity Breakdown Structure. The team will investigate how parallel flows for subsystems (airframe and propulsion) may be modeled in Descartes-Hyperport. The team will use this knowledge in preparation for Phase II of the project which will model development and acquisition costs within the Arena framework for similar case studies as those in Phase I. The team also will perform site visits to obtain better experiential knowledge of turnaround processes. This will include a site visit to Robins Air Force Base (Warner Robins, GA) to examine processes related to their logistics operation. The team has also developed and implemented a simple case study of an operations process within the company. The team developed a DES model of a motorized rover erector set (referred to as Descartes-Rover) and obtained data on the various processes needed to assemble such a machine. Personnel within the firm were timed for several trials in order to obtain distributions on process times. Such simple experiments may aid in the formulation of more complex operational models. The team also hopes to

eventually apply both modeling lessons and Arena-developed models to simulation of other space transportation and infrastructure problems. These could potentially include modeling within a Descartes framework such diverse projects as sub-orbital terrestrial spaceports, International Space Station logistic resupply, and lunar surface human exploration architectures.

SUMMARY

The Descartes framework being developed by SpaceWorks Engineering will develop a robust and detailed simulation to predict turnaround time, recurring ground operations cost, development cost, and acquisition cost for future hypersonic flight vehicles. The simulation will model the individual activities associated with vehicle development and operations using an advanced analytical technique referred to as discrete event simulation (DES). The simulation efforts will focus on the propulsion system of such future hypersonic vehicles, a major operational driver given the highly integrated design of such vehicles. Inputs for such an analysis model will include descriptions of the basic vehicle including concept of operations, mission models, launch/landing facilities, vehicle configuration, performance capabilities, and technologies. The model will be applied to several representative conceptual designs of multi-stage-to-orbit space access vehicles. Descartes will help vehicle designers determine the potential operational benefits of such vehicles, as well as guide our thinking on how one would actually develop and acquire the vehicle itself. The elements in this framework can be adopted for modeling other space transportation or infrastructure concepts in the future.

ACKNOWLEDGMENTS

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

The authors gratefully acknowledge the contributions of colleagues at SpaceWorks Engineering, Inc. (SEI) who contributed to this paper. Mr. Michael Kelly, Operations Engineer in the Engineering Economics Group (EEG) at SEI, is specifically commended for his technical work in developing the Descartes framework and its application to RLV design (Descartes-Hyperport). Mr. William J.D. Escher (Senior Technical Fellow at SEI) and Mr. Carl Ehrlich (SEI Affiliate) all provided sage advice as well as references relevant to operational aspects of past aerospace programs.

REFERENCES

1. Modeling Of Turnaround Operations For Next Generation Space Transportation Architectures, Alex J. Ruiz-Torres, Edgar Zapata, and Marcella L. Cowen, International Journal of Industrial Engineering, 13(1), 18-27, 2006.
2. NASA SP-407, Space Shuttle, ca. 1976; p. 4.
3. Kennedy Space Center, "Space Transportation Ground Processing Operations Modeling and Analysis: A Review of Tools and Techniques," Edgar Zapata, Systems Engineering Office, NASA KSC, June 2003, p. 6.
4. Alemany, K., Street, D.C., Olds, J. R., "LASSO – Lunar Architecture Stochastic Simulator and Optimizer," AIAA 2005-6415, AIAA Modeling and Simulation Technology Conference and Exhibit August 2005, San Francisco, CA.
5. Alemany, Kristina, Comparison of an Expendable and a Reusable Lunar Exploration Architecture Using the Lunar Architecture Stochastic Simulator and Optimizer (LASSO), Space Systems Design Lab, Georgia Institute of Technology, Atlanta, GA, May 2005. [http://www.ssd1.gatech.edu/main/ssdl_pres_archive/KAlemany_MS8900_Sp05.pdf Accessed May 2005.]
6. Cates, G.R. Mollaghasemi, M., "Supporting the vision for space with discrete event simulation," Simulation Conference, 2005 Proceedings of the Winter Simulation Conference, 4-7 Dec. 2005.
7. Cates, G.R. Mollaghasemi, M., et al, "Modeling the Space Shuttle," Simulation Conference, 2002 Proceedings of the Winter Simulation Conference.
8. Reeves Jr., John Daniel, Discrete Event Simulation of Reusable Launch Vehicle Ground Operations (RLVSim), Space Systems Design Lab, Georgia Institute of Technology, Atlanta, GA, July 2004. [http://www.ssd1.gatech.edu/main/ssdl_pres_archive/Reeves_8900_RLVSim.pdf Accessed May 2005.]
9. Zapata, Edgar, GEM-FLO, National Aeronautics and Space Administration, 2003. [http://science.ksc.nasa.gov/shuttle/nexgen/GEM-FLO_main.htm. Accessed 4/7/05.]
10. Rockwell Automation, "Manufacturing," Rockwell Automation Arena Industry Success Stories [online presentation charts], URL: http://www.arenasimulation.com/manufacturing.asp [cited 29 July 2008].
11. Nevins, M. Steele, M., Zapata, E., "SpaceSim – Spaceport Simulation: An Object Oriented Discrete Event Simulation," NASA KSC Next Gen Site [online website], URL: http://science.ksc.nasa.gov/shuttle/nexgen/SpaceSim_main.htm [cited 29 July, 2008].
12. Meucci, J., "Delta Airlines ACS Planning & Performance," ProModel Solutions Conference 2006 [online presentation], September 14, 2006, URL: http://www.promodel.com/solutionsconference/presentations/Delta-Atlanta-Lobby.ppt [cited 29 July 2008].
13. Lufthansa Airlines, "De-peaking Lufthansa Hub Operations at Frankfurt Airport," Rockwell Automation Arena Industry Success Stories [online presentation charts], URL: http://www.arenasimulation.com/pdf/Successstory_SAT-ARENA_Lufthansa.pdf [cited 29 July 2008].
14. Powell, E., "Use of Discrete Event Simulation in New Aircraft Design," SAE Aerospace Manufacturing Technology Conference & Exposition, SAE 1999-01-2269, Bellevue, WA, June 1999.
15. Deborah Sadowski, D. and Bapat, V., "The Arena Product Family: Enterprise Modeling Solutions," Proceedings of the 1999 Winter Simulation Conference.
16. W. David Kelton et al. Simulation with Arena, Fourth Edition. McGraw-Hill, New York, 2007.

17. A Catalog of Spaceport Architectural Elements, A Tool for Assisting Developers of Future Space Transportation Systems, Preliminary Modeling Definition Document, October 1997. URL: <http://science.ksc.nasa.gov/shuttle/nexgen/catalog.htm>, URL: http://science.ksc.nasa.gov/shuttle/nexgen/Nexgen_Downloads/Catalog_ezMar03cleanup.doc

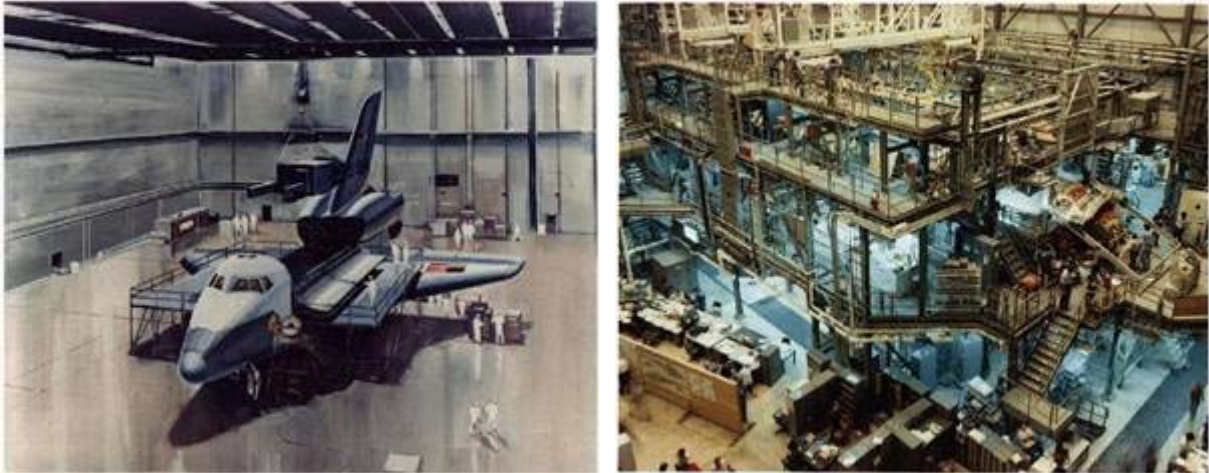
18. DePasquale, D., Charania, A., "I-RaCM: A Fully Integrated Risk and Life Cycle Cost Model," AIAA-2008 7703, Space 2008, San Diego, California, September 9 - September 11, 2008.

19. Personnel Launch System (PLS) Study - Final Report. Rockwell International. NASA Contractor Report 187620, October 1991.

20. Advanced Manned Launch System (AMLS) Study - Final Report. Rockwell International. NASA Contractor Report 189673, September 1992.

21. Douglas Morris, "X-15/HL-20 Operations Support Comparison," NASA TM-4453, June 1993

22. Ted Weber, REal World Analysis of Reliability and Downtime (REWARD) model, URL: <http://www.tfwsciences.com/>.



**Figure 1. Expectation and Reality of Space Shuttle Turnaround Operations
(Orbiter Surrounded by Platforms and Equipment in Right Image)**

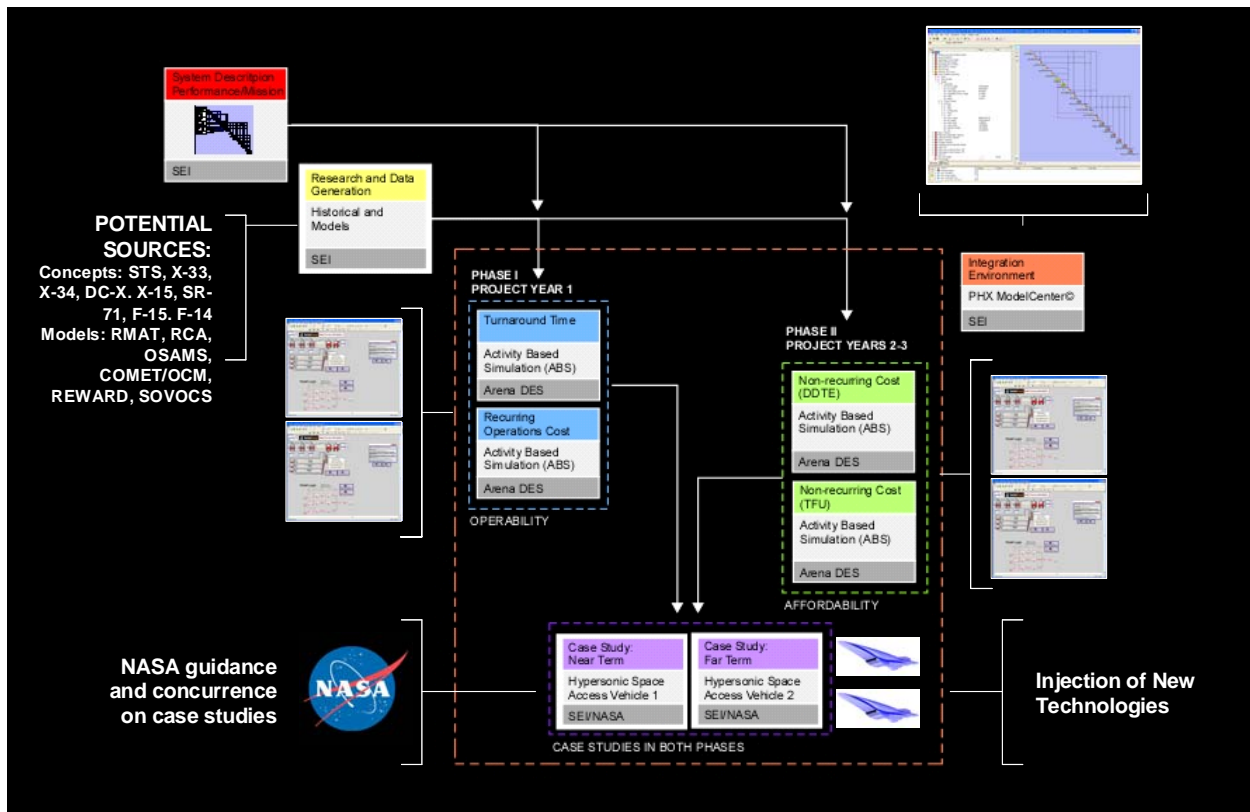


Figure 2. Overall Project Map

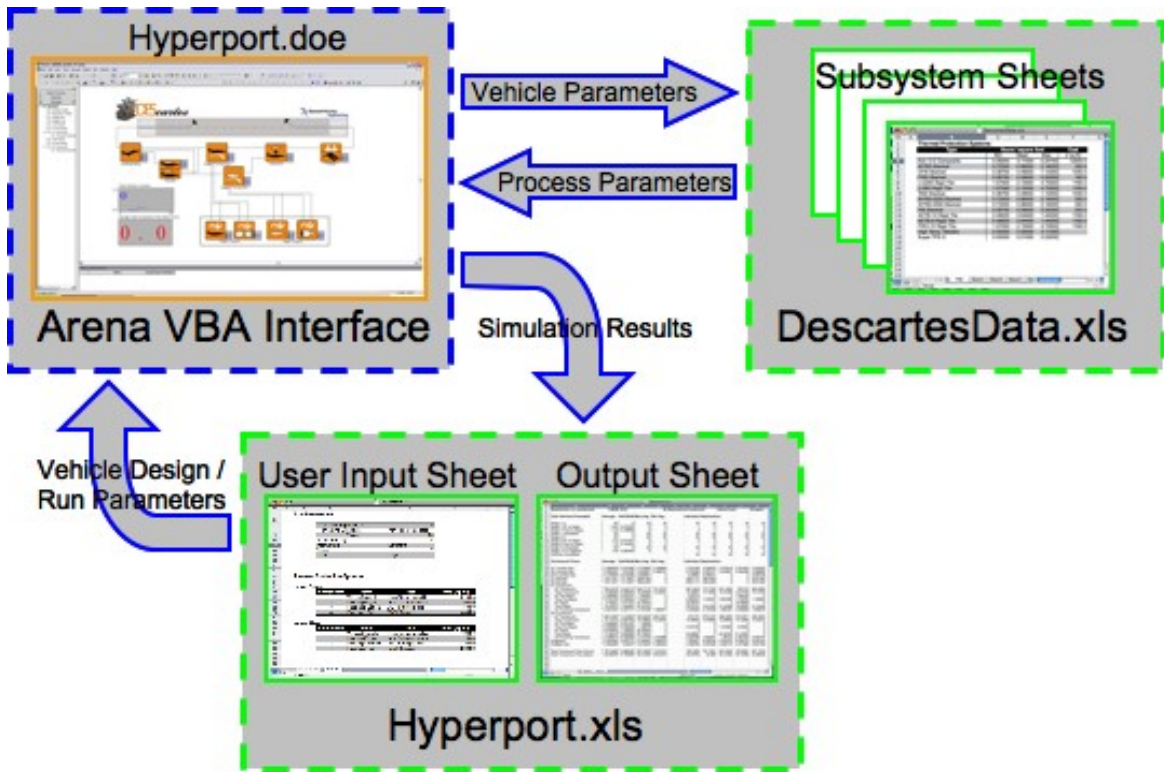


Figure 3. Schematic of Descartes-Hyperport Functionality

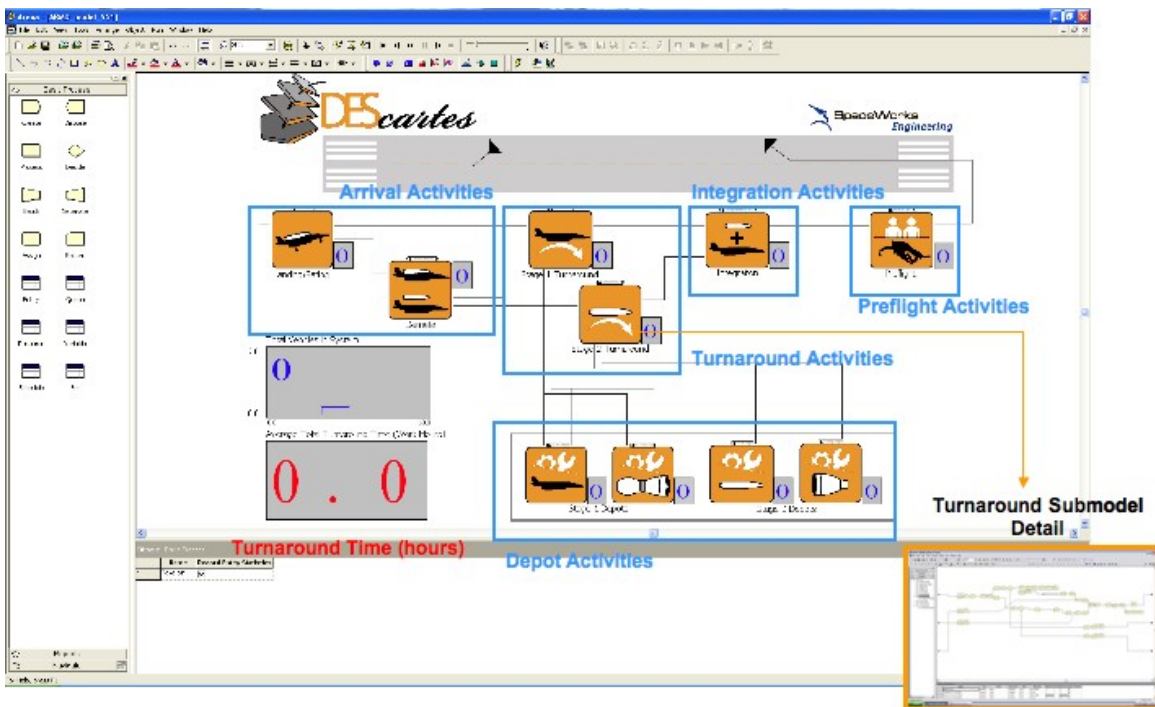
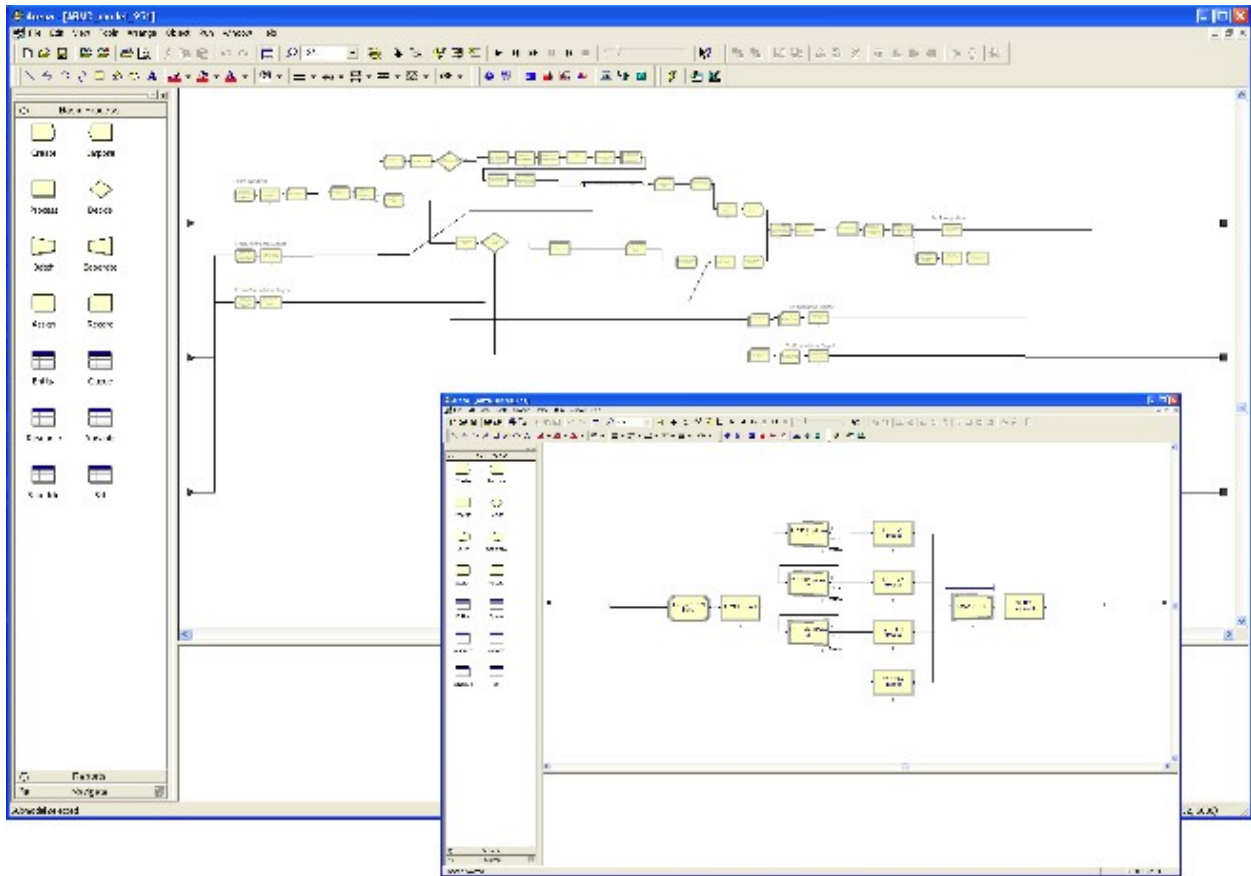


Figure 4. Descartes-Hyperport Screen Example (Arena model Hyperport.doe)



**Figure 5. Descartes-Hyerport Stage 1 Turnaround Facility Submodel
(Inset Shows Stage 1 TPS Turnaround Submodel)**