MAXIMIZING SCIENCE DATA RETURN: A FAST, COST-EFFECTIVE METHOD TO SIMULATE AND ANALYZE EXPECTED DATA RETURN FOR THE MARS RECONNAISSANCE ORBITER MISSION

Cherissa Fischer
Ball Aerospace & Technologies Corp.

Abstract:

The Mars Reconnaissance Orbiter (MRO) will investigate Mars to a level of detail that will generate a 25-fold increase in data compared to previous missions, demanding time on an already crowded Deep Space Network (DSN).

The MRO scientists and spacecraft provider will need to carefully design the spacecraft's instruments, command and data handling (C&DH), and telecommunications subsystems to maximize the science data transmitted back to Earth. However, optimizing the design of each individual subsystem is cost-prohibitive and does not necessarily maximize science data return.

This work outlines an improved methodology; a spacecraft data flow model created using commercial-off-the-shelf software in a fraction of the time of previous approaches with significant schedule and cost savings. The modeling approach presented can be used to provide rapid, cost effective analysis of data flow needs during mission requirements definition, detailed design, assembly, test, launch, and operational phases of a spacecraft's lifetime.

1 Introduction

System design, including trade studies, is an important part of the spacecraft design process. This is especially true, as new missions with higher data rates[1] have to compete with finite DSN resources[2]. The system design process can be aided by the use of computer modeling. A system-level spacecraft data flow model can accurately pinpoint where bottlenecks exist and provide to subsystems engineers specific components on which to concentrate their design efforts.

Discrete Event (DE) modeling, long used in computer and computer network design, presents a great potential for improving spacecraft design. During the design phase of a program, a DE model can simulate the impacts of various design options to quickly perform trade studies. As the spacecraft matures, the DE model can be correlated to reflect actual spacecraft performance using results from routine tests performed on the spacecraft's flight boxes. The verified model can then be used to quickly analyze the impacts of design changes during the build phase. During the operational phase of the program, the spacecraft DE model can assist operations engineers in recovering spacecraft performance after an on-board failure. Finally, it can help mission operations planners and scientists plan out the optimum scheduling of on-board science instruments. DE modeling is a tool that enhances existing practices and improves a spacecraft program's schedule and budget performance throughout the spacecraft life cycle.

DE modeling has recently reached the affordable range for most space programs. Previous applications of DE modeling cost tens of thousands of dollars and could only run very capable number-crunching computers. With the introduction of more powerful desktop computers and the price reduction in commercial-off-the-shelf software tools, a DE model can be quickly created and used to analyze spacecraft performance long before the build phase of a program. This work describes the process used to create a DE model named DataFlow, which was written in less than 60 hours of engineering labor and which can pinpoint specific, relatively inexpensive areas for improvement in spacecraft design.

2 DE Modeling for Spacecraft

2.1 Model Requirements

A useful, inexpensive DE model must achieve multiple requirements. First, it must be easy for users to manipulate so that design changes or trade studies can be modeled immediately. A couple examples of easy to use interfaces include an Excel spreadsheet or a MATLAB® Graphical User Interface (GUI). The model must also adequately simulate the relevant subsystems and constraints on the overall spacecraft design. Most importantly, the DE model must clearly show "Bottleneck" areas such as how long it takes to read an image out the instrument buffer onto the spacecraft bus, how often the spacecraft can downlink information, and the spacecraft on-board memory usage.

2.2 Modeling Process

The process of creating a spacecraft model begins with the inputs to the spacecraft bus. The major inputs to the model are the instrument, spacecraft, and mission characteristics. This includes data rates from the instruments, communications capability of the spacecraft bus, receiver station capability, and receiver station availability. For example, receiver station availability is impacted by any "blackout" times caused by the spacecraft passing behind the planet Mars while communicating to Earth. A typical modeling process used to create a spacecraft DE model is shown in Figure 1.

The final piece of this DE modeling process is the automated interface. It takes the data input into the Excel spreadsheet or MATLAB GUI, passes it to the variables assigned in the simulator, runs the simulation, and outputs the graphs of interest. Also, the automated interface can iteratively change parameters of interest to perform trades studies using batch processing.

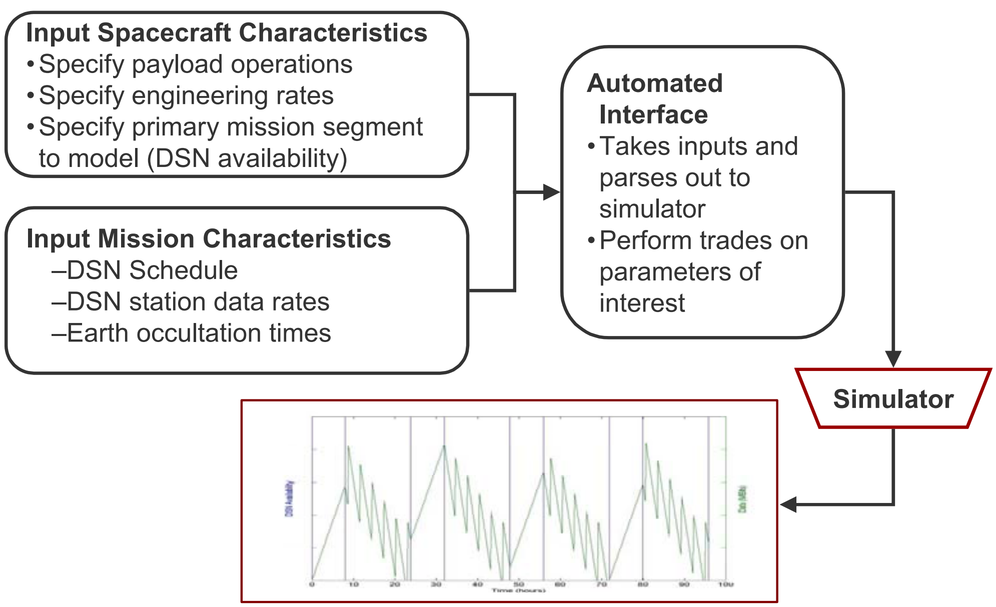

Figure 1: The Spacecraft DE Modeling Process Incorporates the Spacecraft and Mission Characteristics into an Automated Interface for Ease of Use.

3 DataFlow Model Overview

The DataFlow model is described in the following sections. Each section corresponds to a block in Figure 1.

3.1 Simulator

The simulator forms the heart of the modeling process. It is where the spacecraft inputs are processed to simulate spacecraft system performance. The simulator model shown in Figure 2 simulates a picture being taken, a steady stream of incoming measurements, or a combination of the two. Spacecraft subsystem telemetry is added to the data stream, which is then collected and read into the spacecraft memory. Finally, the data is read out of memory and broadcast to Earth during the next DSN pass.

In the Payload Operations area, the frequency at which instruments take data is modeled using a pulse generator or a constant stream of data. If applicable, the outputs of the instruments are converted from pixels to bits of data. Also modeled for most instruments is a description of how fast the data-bus can transfer the instrument output to the spacecraft C&DH system. The Accumulator block contains a piece of code which will output data describing how long it takes to read the simulated image out of the instrument buffer. The Output boxes indicate points at which performance data is recorded for later graphing and analysis.

The combined instrument data is read out to the spacecraft C&DH system. Each instrument has an associated switch so it can be turned on or off from outside the GUI of the model itself.

Spacecraft telemetry is also added to the data stream on the bus. This block's functionality can be expanded to represent state-dependant input from the various spacecraft subsystems as the spacecraft design moves forward in its design process.

All the spacecraft data is stored in the memory block until a DSN window opens up for transmission down to the ground. As the spacecraft design matures, this block expands to account for read-in/read-out speeds.

Another starting point in this model is the telecommunications (telecom) subsystem. Since DataFlow simulates a spacecraft at Mars, the telecom model possesses many mission-unique features. The telecom model originates with the DSN availability. Another aspect of this Direct-to-Earth (DTE) link is the eclipsing that occurs when Mars is between the Earth and the spacecraft. The DTE link also has a range-dependant maximum data rate, which is modeled as a gain on the DSN availability. The data rates are obtained from a look-up table based on the user-input of mission day. Further, the data must be encoded and formatted before being transmitted to Earth, which is modeled as a downgrading of the overall telecom data rate.

The telecom availability is an input to the model of the spacecraft memory storage unit. When a DTE link is open, the model simulates the on-board operations of reading stored data out of memory, data formatting, and transmission to Earth. Finally, all the data received on the ground is accumulated and total mission volume is calculated.

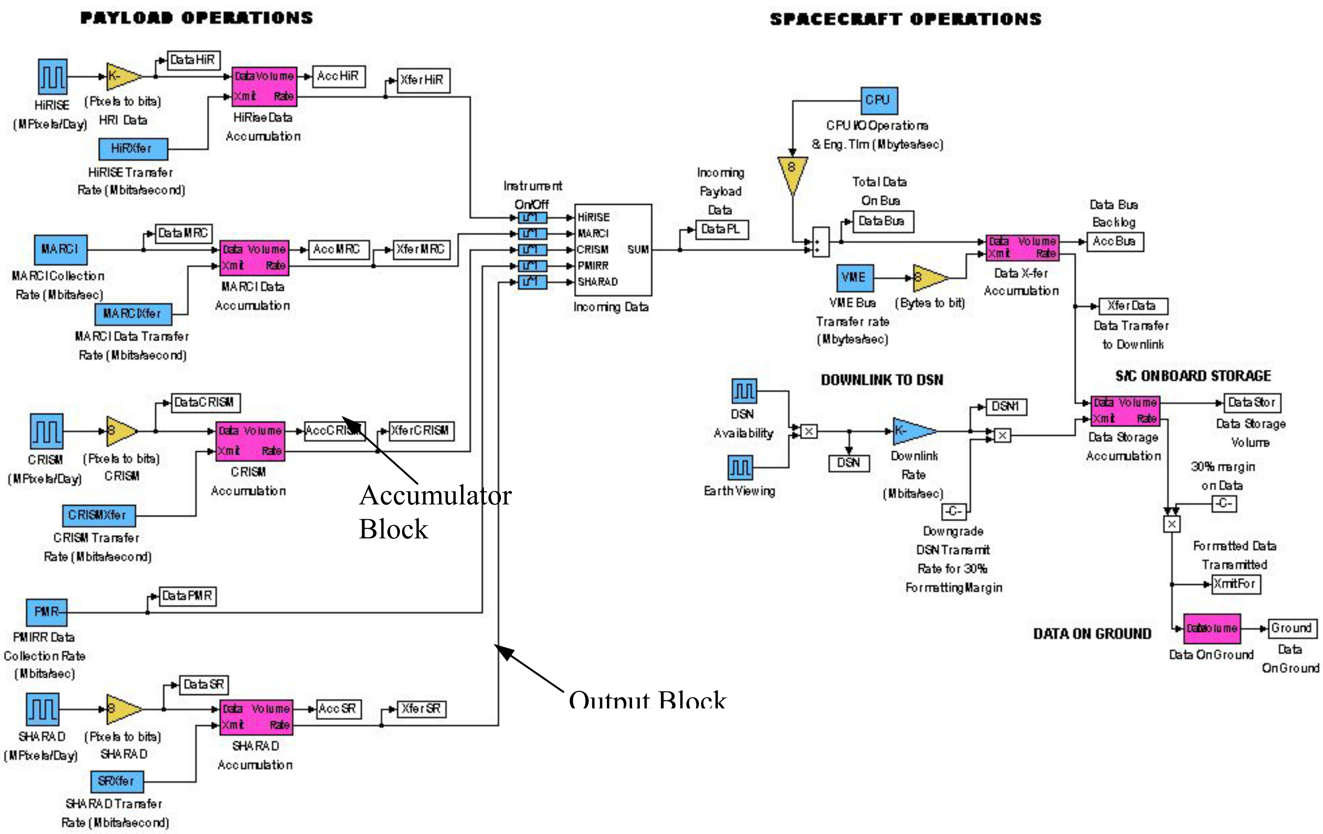

Figure 2: The Spacecraft DataFlow Model as Shown in the MATLAB®/Simulink® GUI.

3.2 Inputting Spacecraft Characteristics

It is important to be able to change the input parameters of the analysis to be able to study the impact of changes to the spacecraft design. As displayed in Figure 3, the various component parameters in the DataFlow simulation are aliased (i.e., they are variables instead of numbers) so that values can be easily and quickly changed from outside the Simulink GUI. The aliasing scheme allows trade studies to be performed on parameters of interest with batch processing. The numerical values corresponding to the DataFlow's aliases are input using a Microsoft® Excel file, shown in Figure 4.

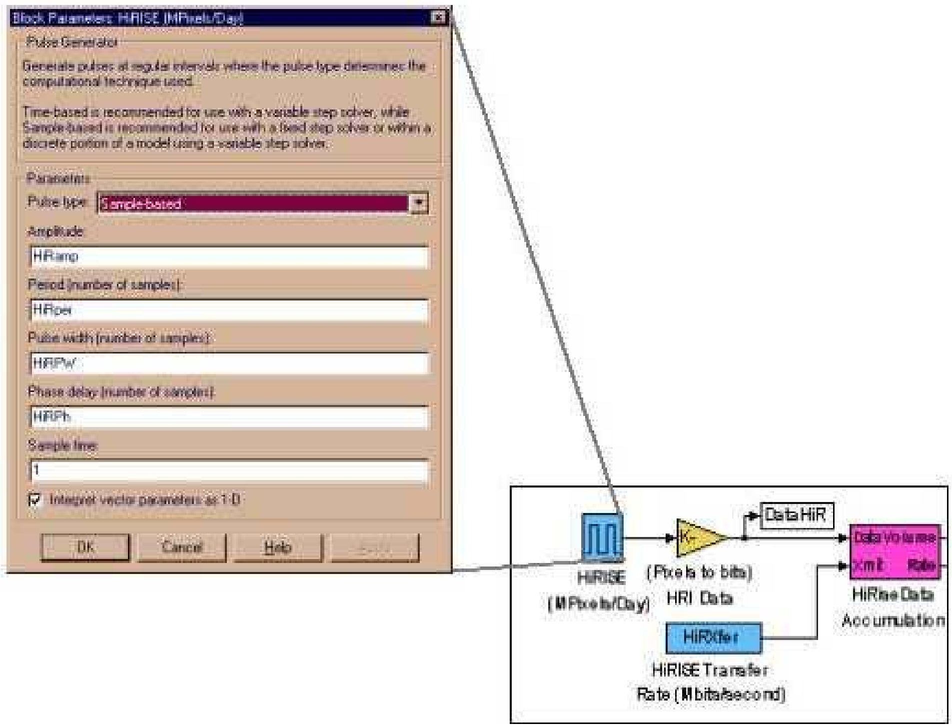

Figure 3: Parameterized Dataflow Simulator Inputs Allow for a Simpler User Interface and the Ability to Automate Trade Studies

The user can use a familiar Excel spreadsheet like the one shown in Figure 4 to input the instrument characteristics as well as those for the spacecraft C&DH and telecommunications subsystems. The input values are passed to automated interface by clicking on the “Save dfi.txt File” button.

The Excel spreadsheet also assists the user in identifying the specific model outputs he or she may be interested in.

Inputs			
Name	**Alias**	**Value**	**Units**
HiRISE (Mpixels/Day)			
Amplitude	HiRamp	###	Mpixels
Period	HiRper	###	days
Pulse Width	HiRPW	1	number of pictures (must be INTEGER)
Phase Delay	HiRPh	###	delay from model start by HRIPh seconds
HRI Transfer Rate (Mbits/sec)			
Constant Value	HiRXfer	###	Mbits/sec
MARCI Collection Rate (Mbits/sec)			
Constant Value	MARCI	###	Mbits/sec
MARCI Data Transfer Rate (Mbits/sec)			
Constant Value	MARCIXfer	###	Mbits/sec
CRISM (Mpixels/Day)			
Amplitude	CRISMamp	###	Mpixels
Period	CRISMper	###	days
Pulse Width	CRISMPW	1	number of pictures
Phase Delay	CRISMPh	###	delay from model start by VNIRPh seconds
CRISM Data Transfer Rate (Mbits/sec)			
Constant Value	CRISMXfer	###	Mbits/sec
PMIRR Data Collection Rate (Mbits/sec)			
Constant Value	PMR	###	Mbits/sec
SHARAD (MPixels/day)			
Amplitude	SRamp	###	Mpixels
Period	SRper	###	days
Pulse Width	SRPW	1	number of pictures
Phase Delay	SRPh	###	delay from model start by SRPh seconds
SHARAD Transfer Rate (Mbits/sec)			
Constant Value	SRXfer	###	Mbits/sec
CPU I/O Operations & Eng. Tlm (Mbytes/sec)			
Constant Value	CPU	###	Mbytes/sec
VME Bus Transfer rate (Mbytes/sec)			
Constant Value	VME	###	Mbytes/sec
Downlink Rate (Mbits/sec)			
Gain	DR	###	Mission Day

Outputs	
Alias	**Meaning**
High Resolution Imager	
DataHiR	HiRISE Data Generated
AccHiR	HiRISE Data Transfer Backlog
XferHiR	HiRISE Data Transfer Rate
MARCI	
DataMRC	MARCI Data Generated
AccMRC	MARCI Data Transfer Backlog
XferMRC	MARCI Data Transfer Rate
CRISM	
DataCRISM	CRISM Data Generated
AccCRISM	CRISM Data Transfer Backlog
XferCRISM	CRISM Data Transfer Rate
PMIRR	
DataPMR	PMIRR Data Generated
SHARAD	
DataSR	SHARAD Data Generated
AccSR	SHARAD Data Transfer Backlog
XferSR	SHARAD Data Transfer Rate
Spacecraft Operations	
DataPL	Incoming Payload Data
DataBus	Total Data on Bus
AccBus	Data Bus Backlog
XferData	Data Transfer to Telecomm
DataStor	Data Storage Volume
DSN	DSN Availability binary yes/no
DSN1	DSN Availability shown as Data rate pulses
XmitFor	Fomatted Data Transmitted
Ground	Data on Ground

Save Excel Workbook: dataInputs.xls

Save dfi.txt File

Figure 4: The Microsoft Excel Interface Assists the User in Changing Input Parameters and Interpreting Output Data.

3.3 Mission Characteristics and Automated Interface

The last two parts of the DataFlow model consist of the mission characteristics and the automated interface. The mission characteristics are listed in a simple text file sorted by mission day. The automated interface is made up of MATLAB script files. The MATLAB scripts fill out the simulation parameters based on the Excel inputs. The scripts also look up the corresponding mission data (available DSN stations, data rates, and Earth occultation times) contained in the mission characteristics text file. Additionally, the scripts can perform trade studies by iterating on certain parameters and re-running the simulation. A single command typed at the command prompt starts the model and outputs various performance graphs.

3.4 DataFlow Output Graphs

Data output from any computer model should be intuitively obvious and easy to for the user to interpret quickly. The DataFlow model supports the user by providing graphical output to meet these needs. Two sample output plots are shown in Figure 5, depicting outputs from the data payload operations area of the simulator. Each spike on the left-hand graph represents a picture being taking by the HiRISE imager. The right-hand graph shows the amount of data stored in the instrument buffer. After the image is taken, it will remain in the instrument buffer until it is read out onto the spacecraft bus. Bus transfer speeds are typically smaller than the image size, so it will take a finite level of time to completely read the image into the spacecraft memory.

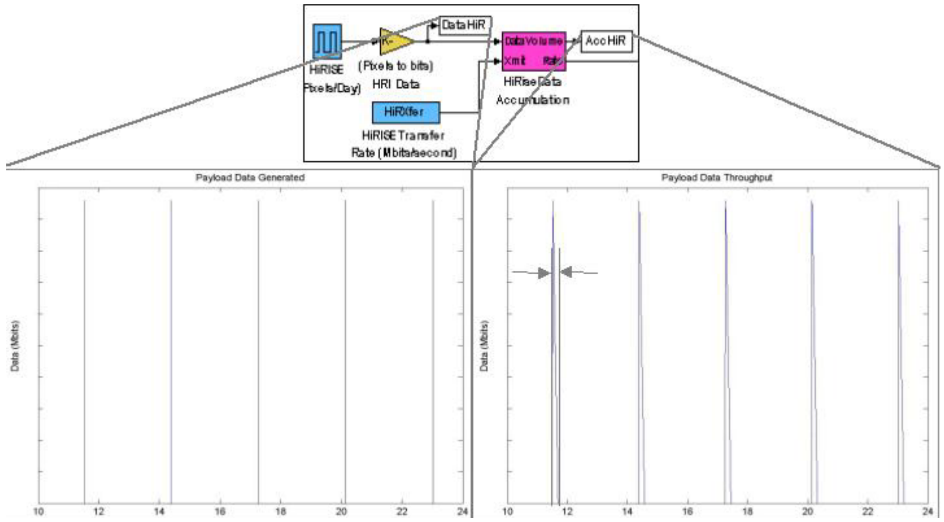

Figure 5: Example Output Graphs from the Payload Operations Area of the Simulator.

The example output in Figure 6 shows the amount of data in the spacecraft on-board memory on the right axis of the graph. The left axis displays status of the DSN link back to Earth. The left and right axes have differing scales. When the DSN link is not open, the incoming instrument data is stored in the spacecraft memory. When the window opens, the data is read out of the memory. In this example, the imagers are still taking pictures, so the memory storage usage will continue to spike periodically as new data comes in. At the end of the DSN window, the memory in storage continues to climb until the next DSN window opens. From the graph, one can see that the complete cycle to zeroing out the memory takes about 3 full days.

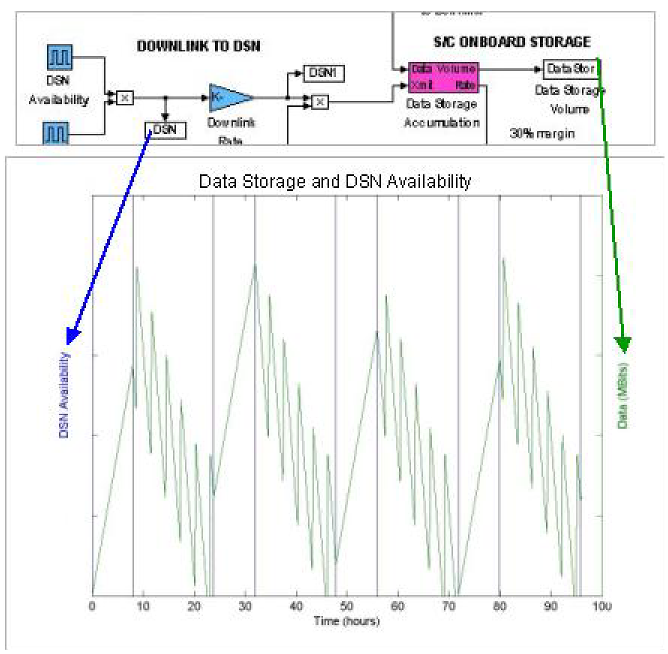

Figure 6: The Data Storage and DSN Availability Graph allows the Spacecraft Analyst to Quickly Determine Whether the Given Spacecraft Design Meets Mission Requirements.

4 Systems Analysis Using DE: An Example Problem

The previous sections showed data from an ideal system design to illustrate generally how to build a spacecraft DE model. Presented below is an example problem that generally describes how the systems analyst and subsystems engineers can apply DE modeling to spacecraft design.

4.1 Problem Statement

In this example, the C&DH subsystem has a given level of spacecraft memory in its design. The telecommunications subsystem designs and mission characteristics mean that data is downlinked to Earth at a given rate. Given this information, the spacecraft system analyst creates a DE model of the spacecraft. The output graphs from this model are used to analyze the spacecraft performance from a *systems* perspective.

In this example problem, the model shows that the spacecraft on-board memory is saturated a certain amount of time into the scenario (top graph), but that the available DTE link is not fully used (bottom graph).

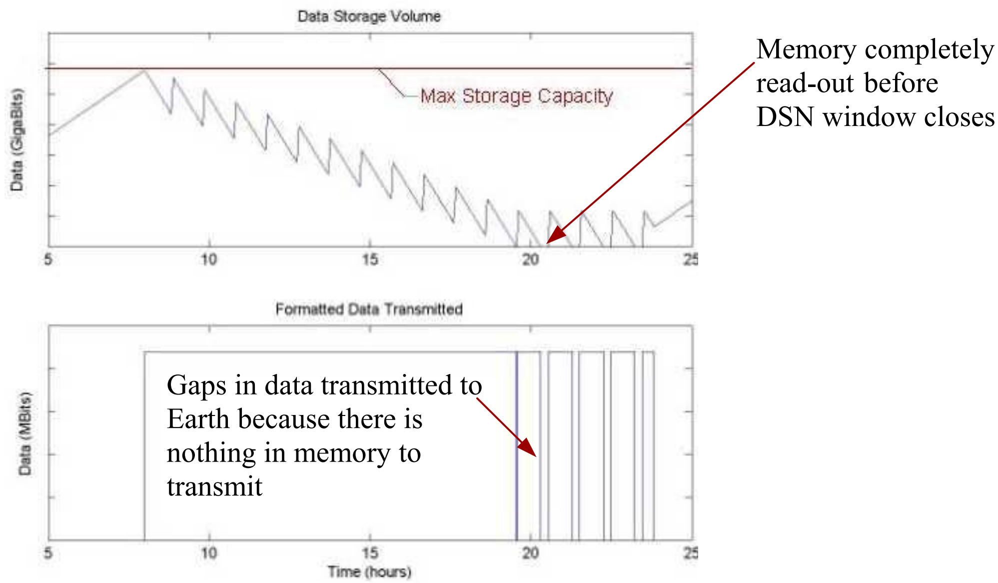

Figure 7: Output Graphs from a DE Model Indicate an Inefficient Design.

4.2 Possible Actions

There are many possible decisions the spacecraft design team may make. For example, extra memory might be added to the spacecraft C&DH box -- a much cheaper solution than upgrading the entire C&DH system. Alternatively, the team could decide that a less capable, less expensive telecom subsystem design is adequate to clear the on-board memory, giving dollars back to the program. Similarly, the team could decide that fewer DSN hours are needed to downlink all of the data.

The DE model allows such trades to take place before the design is built, identifying problems long before the build phase of the program, when they are traditionally discovered.

5 A DE Model Grows Alongside its Spacecraft

A spacecraft DE model such as DataFlow can grow with the spacecraft design, modeling down to the board-level functions if necessary. For example, lower-level functionality can be modeled in Simulink using so-called "subsystem blocks." A subsystem block is a graphical representation of a group of Simulink blocks and operations at a lower level. The Accumulator block in the DataFlow simulator is an example of a subsystem block. It contains a more complex connection of blocks than is reasonable to show at the spacecraft system level. Thus, as the spacecraft design matures, a simple block model of a component can be replaced with a subsystem block to more closely model the actual component functionality, if such level of detail is desired.

6 Summary: Future Direction / Other Applications

Discrete Event modeling represents a large return on investment when designing spacecraft. DE modeling presents a great application to spacecraft design, since models can be created quickly and the software is inexpensive. Once a DE model has been created for one spacecraft, the custom subsystem blocks can be stored in a library of blocks for easy re-use on future spacecraft, reducing model creation time even further.

Furthermore, many optimization tools can be used to iteratively optimize the data bus design. While DataFlow did not utilize any of the available optimization tools, it is expected that future applications will make use of optimization algorithms.

Finally, a model like DataFlow can be expanded to model and optimize ground system operations, such as the flow of science data from the DSN to the various science investigators on a mission. This would ensure that not only is the maximum science data returned to Earth, but is also expedited to the science team members.

Bibliography

1. (2000 December 4). *Deep Space Network Faces Major Crunch.* Space.com [WWW document] URL http://www.space.com/news/spaceagencies/dsn_crunch_001204.html.
2. Leonard David. (2001 July 16). *Zooming In On Mars: The Road to Human Missions.* Space.com [WWW document] URL http://www.space.com/scienceastronomy/solarsystem/mro_spysat_010716-3.html.