SANDIA REPORT
SAND2007-2788
Unlimited Release
Printed May 2007

Network and Adaptive System of Systems Modeling and Analysis

John P. Eddy, Dennis J. Anderson, Craig R. Lawton, and James E. Campbell

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2007-2788
Unlimited Release
Printed May 2007

Network and Adaptive System of Systems Modeling and Analysis

Dr. John P. Eddy, SMTS
System Sustainment & Readiness Technologies Department
P.O. Box 5800
Albuquerque, NM 87185-1176

Dennis J. Anderson, PMTS
System Sustainment & Readiness Technologies Department
P.O. Box 5800
Albuquerque, NM 87185-1176

Craig R. Lawton, PMTS
System Sustainment & Readiness Technologies Department
P.O. Box 5800
Albuquerque, NM 87185-1176

Dr. James E. Campbell, DMTS
System Sustainment & Readiness Technologies Department
P.O. Box 5800
Albuquerque, NM 87185-1176

Abstract

This report documents the results of an LDRD program entitled "Network and Adaptive System of Systems Modeling and Analysis" that was conducted during FY 2005 and FY 2006. The purpose of this study was to determine and implement ways to incorporate network communications modeling into existing System of Systems (SoS) modeling capabilities. Current SoS modeling, particularly for the Future Combat Systems (FCS) program, is conducted under the assumption that communication between the various systems is always possible and occurs instantaneously. A more realistic representation of these communications allows for better, more accurate simulation results. The current approach to meeting this objective has been to use existing capabilities to model network hardware reliability and adding capabilities to use that information to model the impact on the sustainment supply chain and operational availability.

Future work will extend these capabilities to allow for statistical treatment of non-hardware related network failures.

Acknowledgements

The Network and Adaptive System of Systems Modeling and Analysis LDRD program team would like to acknowledge the significant support, time, and effort provided to the program by Robert Cranwell, LDRD Program Manager. The team also acknowledges the support of and guidance from the members of the Modeling and Simulation Thrust of the Emerging Threats Investment Area: Russ Skocypek, Alan Nanco, John Wagner, Robert Cranwell, and Ron Trellue. Finally, the team acknowledges and thanks Chris Atcitty for his contributions to the program.

Table of Contents

Abstract ... 3
Acknowledgements ... 5
Table of Contents ... 6
List of Figures ... 7
List of Tables ... 8
Executive Summary ... 9
1. Introduction ... 11
1.1. Problem Background ... 11
1.2. Goals and Objectives of the Project ... 12
2. Network Modeling ... 13
2.1. Systems of Systems Application Toolkit (SoSAT) ... 13
2.1.1. Functions and Elements ... 13
2.1.2. Supply Networks ... 14
2.2. Network Simulation ... 14
2.3. Solution Steps ... 16
2.3.1. Information Gathering ... 16
2.3.2. Code Modifications ... 17
2.3.3. Data Input to Existing Simulation ... 17
2.3.4. Reference State Input ... 17
2.3.5. System Function Definition ... 26
3. Example Problem ... 28
3.1. Problem Description ... 28
3.2. Simulation Input ... 30
3.3. Results ... 45
4. Conclusions and Future Work ... 51

List of Figures

Figure 2-1. Simple Black Network Function Block Diagram. 15
Figure 2-2. Example Reference Element Specification. 16
Figure 2-3. Improved Reference Assignment Form. 18
Figure 2-4. Dialog for Final Filtered Node User Selection. 19
Figure 2-5. Node Type Filtering Options. 20
Figure 2-6. Referring System Tree System Node Popup Menu. 21
Figure 2-7. Referring System Tree Non-System Node Popup Menu. 22
Figure 2-8. Reference State Item Popup Menu. 22
Figure 2-9. Reference State List Popup Menu. 23
Figure 2-10. The New Reference State Creation Dialog. 23
Figure 2-11. Referred System Tree Function Node Popup Menu. 24
Figure 2-12. Referred System Tree System Node Popup Menu. 24
Figure 2-13. Incomplete Reference State Specification. 25
Figure 2-14. Referenced Functions List Pop Up Menu. 25
Figure 2-15. Simple Circular Reference. 26
Figure 2-16. Multi-layer Circular Reference 26
Figure 2-17. Example Circular Reference Detection Error Dialog. 27
Figure 2-18. Circular Reference Path for Example Problem. 27
Figure 3-1. Input of Simulation Parameters for the Network Example Problem. 30
Figure 3-2. Input of the Part Definitions for the Network Example Problem. 31
Figure 3-3. Input of the Part Groupings for the Network Example Problem. 31
Figure 3-4. Input of the Consumable Definitions for the Network Example Problem. 32
Figure 3-5. Input of the Consumable Groupings for the Network Example Problem. 33
Figure 3-6. Input of the Service Definitions for the Network Example Problem. 33
Figure 3-7. Input of the Provider Services for the Network Example. 34
Figure 3-8. Input of the System Types for the Network Example. 35
Figure 3-9. Input of the Primary Elements for the Network Example. 36
Figure 3-10. Input of the Primary Elements Without Spares for the Network Example. 36
Figure 3-11. Input of Consumables Required by Platform Types in the Network Example. 37
Figure 3-12. Input of the Platform Functions of the Network Example. 38
Figure 3-13. Input of the Scenario Definitions of the Network Example. 39
Figure 3-14. Input of the Platform Instance Counts of the Network Example. 39
Figure 3-15. Input of the Force Structure of the Network Example. 40
Figure 3-16. Input of the System Assignments of the Network Example. 41
Figure 3-17. Input of the Spare Inventory Assignments of the Network Example. 41
Figure 3-18. Input of the Consumable Inventory Assignments of the Network Example. 42
Figure 3-19. Input of the Service Assignments of the Network Example. 42
Figure 3-20. Input of the Scenario Assignments of the Network Example. 43
Figure 3-21. Input of the Supply Connections of the Network Example. 43
Figure 3-22. Input of the External References of the Network Example. 44
Figure 3-23. Input of the System Function Modifications of the Network Example. 45
Figure 3-24. Demonstration of *ineffective* component failure overlap 48
Figure 3-25. Demonstration of *effective* component failure overlap 48
Figure 3-26. Effect of Decreased Network Hardware Reliability on A_o. 50

List of Tables

Table 3-1. Example Problem Platforms 28
Table 3-2. Example Failure Rate Calculation for E1-E6 of C2V. 29
Table 3-3. Observed Effects of the Network with Notional but Representative Data................. 47
Table 3-4. Comparison of Perfect with 100% Imperfect Network Hardware. 49

Executive Summary

Evaluating the effects of imperfect network communication on mission effectiveness is an immediate need for military systems such as the Future Combat System (FCS). This effort is in support of the more general System of Systems (SoS) modeling capability being developed called the System of Systems Analysis Toolkit (SoSAT). For more information on SoSAT, see SAND2005-0020.

The goal of the Network and Adaptive System of Systems Modeling and Analysis LDRD was to add the ability to account for the effects of imperfect network communications on mission effectiveness into SoSAT. Network communications are unlike other interactions modeled in SoSAT in that they do not rely solely on hardware operability. While hardware is a factor, there are also a number of other factors that can lead to network outages such as bandwidth limitations and environmental conditions.

To achieve the goals of this LDRD, the details of the FCS network were researched for the purpose of developing reliability models of the relevant hardware, mimicking the organization of systems into networks and subnetworks, and discovering the different waveforms used for inter-platform communication. The majority of this information was provided by a point-of-contact for FCS C4ISR RAM-T.

As a result of this research, SoSAT now has the ability to require network availability when forming logistics relationships such as those needed for consumables resupply, spare part ordering and delivery, and contracting of services. This document includes details of an example of a notional FCS BCT with a communications network along with results and discussion. The new capabilities given SoSAT are shown to perform as expected and a considerable list of future tasks and research opportunities is presented as a result.

1. Introduction

This section begins with a brief description of the problem domain and concludes with a discussion of the desired achievements of the project.

1.1. Problem Background

The focus of the work completed to date has been on the Future Combat Systems (FCS) network and the FCS program has been the primary customer envisioned to have interest in this new capability. As such, the problem is to represent this network in such a way that it can be integrated into the existing System of Systems (SoS) Analysis Toolset (SoSAT) that is being used for FCS system-of-systems modeling and analysis, as will be described in more detail in Section 2.1.

Based on the latest information received, the platforms that make up nodes in the FCS network are organized into subnetworks according to geographical location. All inter-platform communication requires the use of one of a number of possible waveforms. Within a subnetwork, the primary communication waveform is the Local Wideband Network Waveform (WNW). All network capable platforms are capable of communicating using the Local WNW.

Communication between platforms in different subnetworks must take place using one of the so-called “backbone” waveforms. The primary backbone waveform is the Global WNW. This waveform is carried only by specific platforms within a subnetwork and is transmitted directly from one such platform to another with no intermediate relaying. The Network Data Link (NDL) is a high capacity link for bandwidth intensive communications such as video and can be used to offload traffic from the Global WNW in times of high demand. This waveform is carried only by specific platforms within a subnetwork and is relayed through an Unmanned Aerial Vehicle (UAV). The forth waveform considered is the Warfighter Information Network – Tactical waveform (WIN-T). WIN-T is primarily for voice traffic but like the NDL, it may be used to offload traffic from the other backbone waveforms. It is communicated via a satellite. Figure 1-1 shows this organization conceptually.

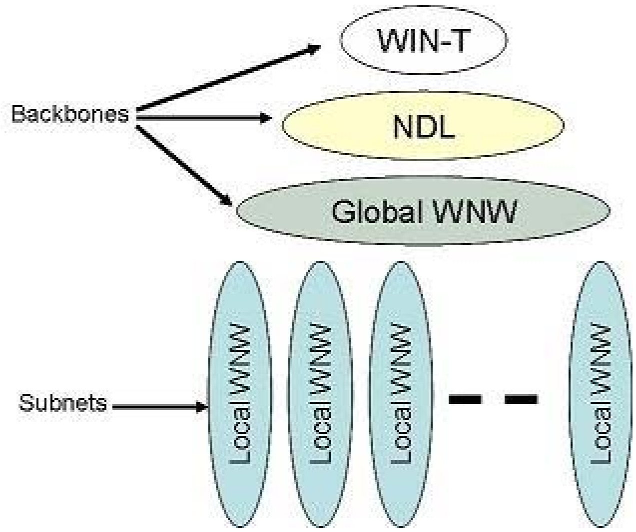

Figure 1-1. FCS Network Overview.

The platforms that use these waveforms to communicate require appropriate hardware. In particular, radio systems have been designed that allow for transmission, reception, and processing of each of these signals. Reliability data for these radios and the components that comprise them has been obtained and is periodically updated. SoSAT is currently well able to make use of this information to provide various output statistics regarding the operational availability (A_o) of the radios themselves. Given the existing capability, this portion of the problem definition is primarily a job of data collection and entry.

The required extension to SoSAT is the introduction of an integrated network layer through which all inter-platform communications must pass. This provides SoSAT with the ability to account for the various network failure modes during execution. The effects of these network issues will be incorporated into the individual platform output statistics.

1.2. Goals and Objectives of the Project

The goal of this project has been to create critical capabilities that are required to meet complex SoS assessment needs in the area of modeling complex, adaptive communications networks.

In support of this, a secondary goal has been to develop algorithms and methodologies and integrate them SoSAT and to further test these algorithms and methodologies by application to a notional FCS Brigade Combat Team (BCT) communications network.

2. Network Modeling

SoSAT operation is based on statistical analysis of platform component reliability with additional capabilities for treating things such as combat damage and the effects of external elements among others. In accordance with this, focus has been placed on developing a statistical treatment of the network including not only the reliability of related hardware but also the effects of other network specific failure modes. Because the existing SoSAT methodology is used as a basis for the network modeling, the next section provides some relevant background on its operation. For more detail, see SAND2005-0020.

2.1. Systems of Systems Application Toolkit (SoSAT)

SoSAT uses a multi-system discrete time simulation for analyses. In this simulation environment, individual systems, also referred to as platforms, such as ground vehicles, aircraft, command and control (C2) vehicles, etc. are represented by State Model Objects (SMOs). The State Model Object allows for the encapsulation of the properties and behaviors of the platforms both conceptually and in actual software implementation. The controlling software provides the environment in which these systems operate and thus manages information external to any one system such as environmental conditions, terrain, supply network information, etc.

2.1.1. Functions and Elements

As mentioned, the SMOs in SoSAT have properties and capabilities. Of particular importance to this work is the ability of an SMO to have functions associated with it. These functions are sometimes referred to as measures of effectiveness. They describe the capabilities and various operations for which a system is intended and may include things like mobility, lethality, sensing, etc. These functions are modeled using equations comprised of elements of various types including:

- Primary Elements – These should be considered the elements that are subject to normal reliability processes such as failures and repairs. Primary elements might be components, line replaceable units, etc.
- Consumable Elements – These elements represent the state of a consumable used by the system during its operation. Examples of consumables might be fuel, water, ammunition, etc.
- External Elements – These are elements that represent influences from outside a system that can affect a system's functionality. Examples of such influences might be a sandstorm or heavy forestation.
- Reference Elements – These are elements that refer to functions of other systems that the current system may require for its functionality. As an example, a Non-Line of Sight (NLOS) cannon's lethality function may have an element that references the targeting function of a UAV.

Each of these elements can be considered a Boolean state variable and will evaluate to either true or false at any given time step in the simulation. They can be combined in various ways to form the aforementioned functions which are Boolean equations. The elements that are of importance to the current network modeling capability are the Primary and Reference element types as will become clear in Section 2.2.

2.1.2. Supply Networks

Another feature of the current simulation important to network modeling is its support of a supply network representation (not to be confused with the communications network which is the topic of this report). The is important because in order for the network failures to affect logistics and ultimately operational availability (A_o), the supply connections of the supply network must be made to depend on the availability of a network path between the customer and the supplier. In the current simulation, such network paths are assumed to be present with 100% surety. Without these network paths, orders for spares, consumable resupply, etc. cannot be placed.

In the current simulation, this is the means by which the communications network will have its most profound effect on the individual systems and their ability to operate. In the future, additional capabilities that will depend on the communications network are planned for SoSAT such as the distribution of Command and Control (C2) information, Common Operating Picture (COP) updates, etc.

2.2. Network Simulation

The network representation designed for this LDRD is based on information obtained from the rapidly evolving FCS program. In the FCS program, there are two primary communication network types. The first is intra-platform communication whereby the various computers and other devices on a platform communicate with one another. This communication is transmitted by wire and the information is typically unencrypted. Because of this, it is considered a secure or classified network and is referred to as the Red Network. The second is inter-platform communication whereby information is transmitted through the air from one platform to another. This is considered an insecure network and thus all sensitive information being transmitted must be encrypted. This is thus called the Black Network. For information to be transmitted from the Red Network of one platform out to another platform, it must pass through an encryption layer before it can enter the Black Network for transmission.

The Red Network is implemented entirely with hardware and so does not require any extension to the existing modeling capabilities. Because of this, all further discussion will focus on the challenges associated with representing the inter-platform communications (Black) network. From here on, references to “the network” will be understood to be references to the Black Network.

The hardware associated with the communications network is housed on the individual platforms. Because of this, it is sensible to have communication functions on these platforms that are comprised of elements relating to this network hardware. As a simple example, consider the case where a Command and Control Vehicle (C2V) has a radio with the hardware elements as shown in the list below.

- PE1 – Mount
- PE2 - Universal Transceiver #1
- PE3 - Universal Transceiver #2
- PE4 - Local Control Display Device
- PE5 - Ground Vehicle Mount
- PE6 - Antenna
- PE7 - Cables
- PE8 - Networking INFOSEC Unit
- PE9 - Wide Band Power Amplifier

Each of these pieces of hardware will be represented in equations using a primary element and thus they are denoted PE#. With this information the following equation could be used to describe the C2V's ability to transmit and receive information on the Black Network.

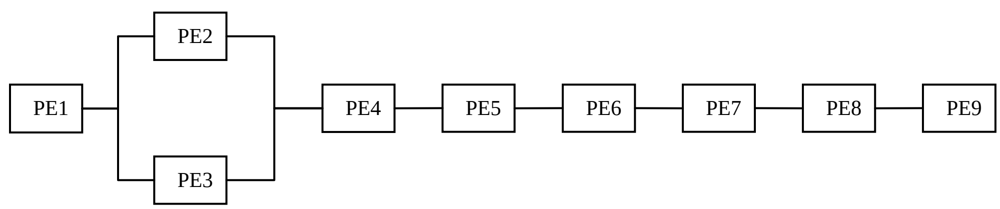

Figure 2-1. Simple Black Network Function Block Diagram.

The above figure can be equally represented as in Equation (2-1 below.

$$PE1 \,\&\, (PE2 \,|\, PE3) \,\&\, PE4 \,\&\, PE5 \,\&\, PE6 \,\&\, PE7 \,\&\, PE8 \,\&\, PE9 \qquad \textbf{(2-1)}$$

In this equation, the "&" represents the logical AND operator and the "|" represents the logical OR operator.

Similar equations will exist for all network capable platforms. The network itself need not be thought of as a physical system but can instead be thought of as an abstract system whose operational availability depends entirely on the functionality of the individual platforms serving as network nodes. In this paradigm, the reference elements as described in Section 2.1.1 are of central importance. Because of this it is necessary to discuss in more detail how the reference elements work and how they are used.

As mentioned above, reference elements are a means by which a function of one platform can depend on functions of other platforms. Reference elements are comprised of a list of functions of other platforms and a count of how many of those functions must be in an operable state in order for the reference element to evaluate to true. Consider Figure 2-2 below for a notional BCT comprised of the following platforms.

- 2 – C2V
- 1 – M1
- 1 – UAV
- 2 – HEMTT
- 1 – ICV
- 1 – MV-E

$$RE1 = \left. \begin{matrix} C2V-001 \backslash \text{Black Network} \\ C2V-002 \backslash \text{Black Network} \\ M1-001 \backslash \text{Black Network} \\ UAV-001 \backslash \text{Black Network} \\ HEMTT-001 \backslash \text{Black Network} \\ HEMTT-002 \backslash \text{Black Network} \\ ICV-001 \backslash \text{Black Network} \\ MV-E-001 \backslash \text{Black Network} \end{matrix} \right|_5$$

Figure 2-2. Example Reference Element Specification.

This figure states that reference element RE1 refers to the Black Network functions of all 8 platforms in our notional BCT and that it requires 5 of them to be operational at any time in order to evaluate to true.

Using these types of reference elements, networks and sub-networks can be defined with functions that mirror the communications functions of the various platforms. Output for the network systems themselves would then include a notion of A_o and all the associated statistics computed for any other platforms.

The example above demonstrates the entirety of the Black Network functionality wrapped up into a single function called Black Network. This need not be the case however. For models requiring more detailed analysis, the functionality of the Black Network can be broken up into a more fine grained representation accounting for the various waveforms.

2.3. Solution Steps

This section details the steps taken to fulfill the requirements of this LDRD.

2.3.1. Information Gathering

The first tasks completed for this project involved the gathering of information about the various aspects of the FCS network. Because the FCS network is still under development, the information required to complete this LDRD has been difficult to obtain and has been subject to frequent change. In order to model the network in SoSAT, two primary pieces of information were required; how the network is organized and what hardware is used to perform communications.

FCS Network Organization

This information describes the topology of the communications network used by FCS as well as its inter-workings. The primary source of this information has been the FCS Command, Control, Communication, Computer, Intelligence, Surveillance, and Reconnaissance (C4ISR) Reliability Availability Maintainability - Testing (RAM-T) point-of-contact (POC).

The result of this information collection effort is described in Section 1.1. The network is an overall system comprised of multiple subnetworks. The subnetworks contain actual network nodes which are the various FCS platforms. Different waveforms are used for communication depending on the relative locations of the nodes that wish to message one another, their capabilities, and the type of information that is being communicated.

FCS Network Hardware

This information describes the actual hardware that exists on each platform for the purpose of inter-platform communication. The primary equipment used by all platforms is the Joint Tactical Radio System (JTRS) radio. Information describing the various internal components of the various JTRS radio configurations used on the various platforms was obtained through the same FCS C4ISR RAM-T POC used to obtain the network organization information. This information is used in the example problem described in Chapter 3.

2.3.2. Code Modifications

In order for the network to be integrated to the level required, some modification of the SoSAT code was required. In particular, the instructions that specify the requirement of network connectivity when forming supply connections had to be devised and inserted. Certain conditions were already present in the code for the formation of these connections including:

- the customer must be in a state in which the required service can be performed,
- the customer must be at a location in which the required service can be performed,
- the supplier must be in a state in which it can perform the required service, and
- the supplier must be at a location in which it can perform the required service

The requirement that both the customer and supplier have an operational network function was added to the list in order that the two may communicate with one another. Prior to this change, the simulation effectively assumed perfect communication capabilities.

2.3.3. Data Input to Existing Simulation

Even without code modification, SoSAT has the ability to accept all of the data collected and to use it in standard reliability analyses. This includes the definition of the force structure, creation of the network system type, entry of MTBF data, etc. With the code modifications described in Section 2.3.2, this information is given special attention when forming supply connections.

In addition to the reliability data and as mentioned in Section 2.2, an actual Network system will be created whose functions consist solely of references to the network functions of the various platforms. The next section describes the input procedure for these references.

2.3.4. Reference State Input

As described in Section 2.2, the network systems are modeled as pure reference systems. In our example of a notional FCS communications network, we will have a single Network platform which, in addition to the required “Operability” function, will have a single function entitled “Black Network”. This single function will be comprised of a single reference element that references the Black Network functions of all network capable nodes in the BCT and will require that some percentage of them be operational in order to be functional.

In order to create large scale pure reference systems such as this, it was necessary to improve the reference assignment capability of SoSAT to allow for large scale, bulk reference assignments based on complex queries and node type filtering. To support this, a new input form has been created to replace the existing form for reference assignment. This new form is shown in Figure 2-3 below.

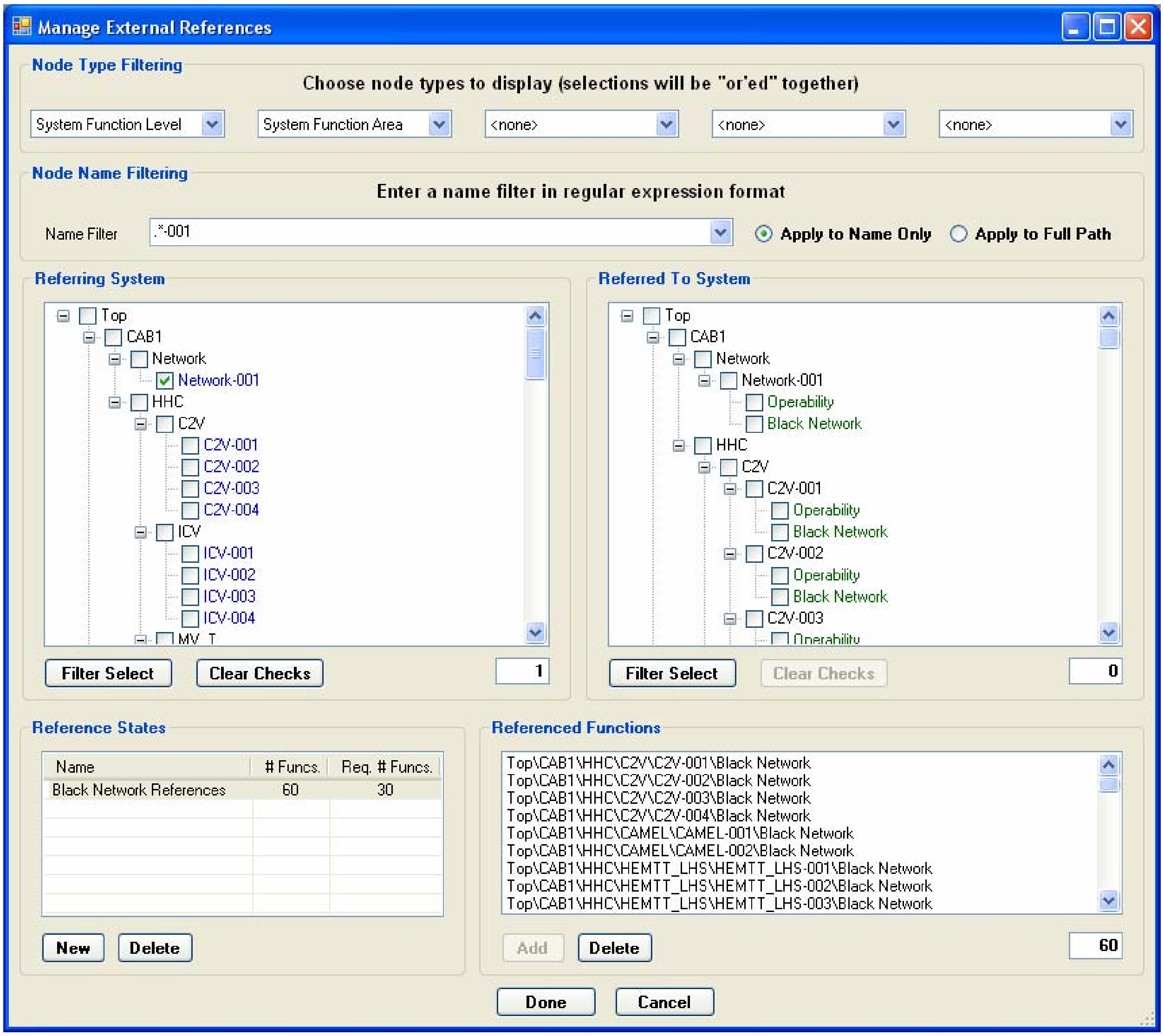

Figure 2-3. Improved Reference Assignment Form.

The purpose of this form is to allow users to select a system or systems to which to add a Reference Element (or Reference State) and to choose which functions of other systems to refer to with this reference element. A system to which a reference element is being added is called a referring system and a system that contains functions that are made part of a reference element is called the referred to system.

To support large scale assignments, this form supports multiple selections in both the referring and referred to tree views. Therefore reference states can be created for multiple systems at the same time and likewise, multiple functions can be added to a selected reference state for all selected referring systems in a single operation. Multiple selections can be made by simply checking as many check boxes in the tree view as desired.

Filtering is performed in a three step process. First, the subset of the nodes in the appropriate tree view is created using the node type filter. That subset is then passed through the name filter creating a reduced subset. Any nodes that pass both filters are presented to the user for final selection in the dialog shown in Figure 2-4 below. The filters can be applied to the nodes of either the referring or referred to trees. The tree to which the operation will apply depends on which *"Filter Select"* button has been pressed.

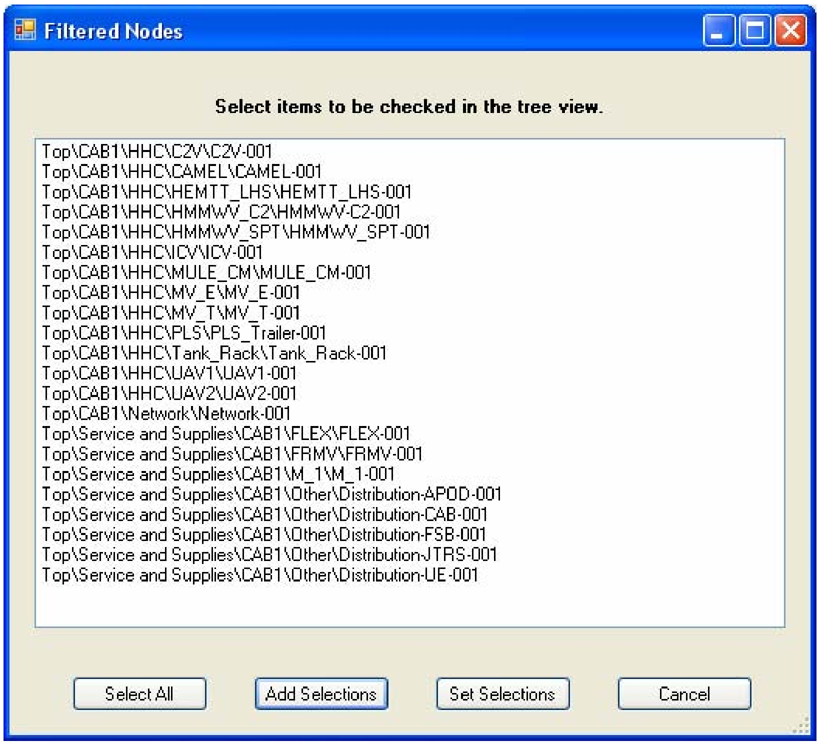

Figure 2-4. Dialog for Final Filtered Node User Selection.

The *"Select All"* button will highlight all the items in the list in Figure 2-4. The *"Add Selections"* button will place checks next to the tree view items associated with the highlighted items without disturbing any other tree view nodes that are already checked. It will also dismiss this dialog. The *"Set Selections"* button will clear all checked check boxes of all nodes in the associated tree and then check only those nodes associated with highlighted items in the list. The result is that the only checked items in the tree will be those that were chosen in the list. This button also dismisses this dialog. The *"Cancel"* button gives the user an opportunity to cancel the selection operation without altering the checks on the tree view. Pressing it will dismiss this dialog without taking any further action.

There are 6 sections in this form. Below is a detailed description of each. Once they are all described, the final section describes how to use them in conjunction with one another to create reference states. They are:

- The Type Filtering Section;
- The Name Filtering Section;
- The Referring System Section;
- The Reference States Section;
- The Referred System Section; and
- The Referenced Functions Section.

The Type Filtering Section

The type filtering section allows a user to specify what types of nodes can be returned by a query. The node types that may be selected are shown in Figure 2-5 below. As can be seen in Figure 2-3, there are 5 node type selection boxes and thus multiple node type selections can be made at the same time. If this is done, then the selections are "or'ed" together meaning that any node of any of the chosen types will pass the filter.

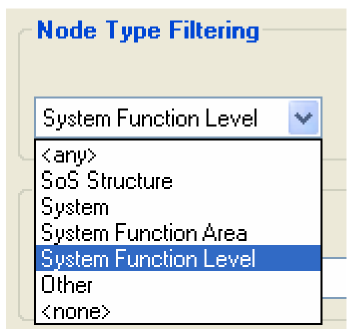

Figure 2-5. Node Type Filtering Options.

The types shown in the figure correspond to the types available for creation when defining the force structure. In addition, there are two special types demarked with angle brackets. They are the <any> and <none> types. Choosing the <any> type indicates that any node type will pass the filter. If this is selected in any of the boxes, then any node type will be allowed regardless of what is in the other boxes. The <none> type means that no type is selected in that particular box. If all boxes have the <none> selection, then no node types will pass the filter and queries will therefore return no results. By default, the first box has the <any> selection and the rest have the <none> selection so all node types will pass. The selections shown in Figure 2-3 will allow all system functions (both levels and areas) to be returned from queries.

The Name Filtering Section

The name filtering section allows a user to specify a filter for the names of the returned nodes in full regular expression syntax. The regular expression syntax is that of the Microsoft .NET framework v2.0 and is Perl 5 compatible. This section also allows a user to specify whether to apply the filter to the leaf node name only or the entire node path. The leaf node name is what is shown next to any of the check boxes in the tree. The full path to a node is build from the name of the node and the names of all parents of the node. So for example, consider the node named *Network-001* in Figure 2-3 above. The full path to that node is *Top\CAB1\Network\Network-001*. Given this, an expression that could be used to find all Network systems in CAB1 might look something like:

$$.*CAB1\backslash Network.* \tag{2-2}$$

when applied to the full paths. The field into which the name filter is entered is actually a combo box that keeps a record of all search strings for easy reuse. This list will remain in tact throughout an entire run of the program meaning that the entries are not lost simply because a user navigates to a different form and then back again.

There is a special default entry for this field that reads <any>. This entry is equivalent to the regular expression ".*"and will thus match any name.

The Referring System Section

This section houses the referring systems tree view. It is only sensible to select system nodes in this tree. They are the only node types to which reference states can be added. You will notice however that there are check boxes next to all nodes, not just the system nodes. Selection of any of the parent nodes results in the selection of all child nodes. This feature makes it simple to select all systems in a particular branch of the force structure. Nodes checked in this tree are the targets of any reference state creation.

There are operations that can be taken from within the referring systems tree view. For example, right clicking on a system node will bring up the menu shown in Figure 2-6 below.

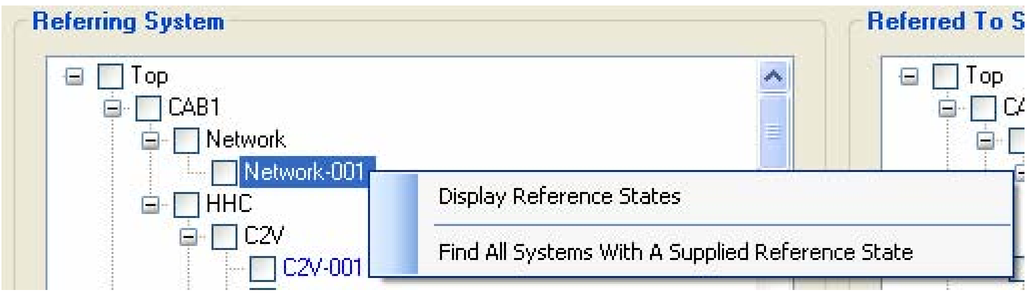

Figure 2-6. Referring System Tree System Node Popup Menu.

The *"Display Reference States"* option brings up a dialog window showing the details of all reference states of the chosen system. The *"Find All Systems With A Supplied Reference State"* option will bring up a dialog into which a user can enter the name of a reference state. This will cause a filtering operation and all nodes that have a reference state by the supplied name will be presented for further selection.

Right clicking on a non-system level node or in an area not associated with a node will result in the popup menu shown in Figure 2-7 below where the entry has the same meaning as above.

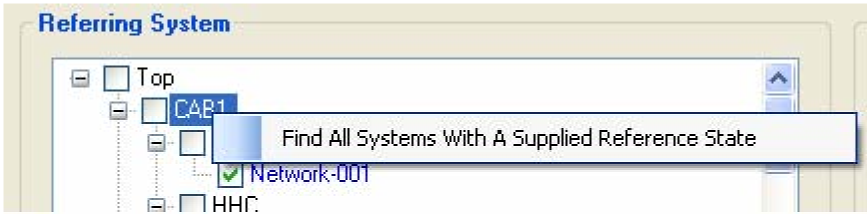

Figure 2-7. Referring System Tree Non-System Node Popup Menu.

The Reference States Section

The reference states section shows all reference states that are common to all selected referring system nodes. So for example, if you have chosen 10 nodes in the referring systems tree by checking them and each of them has a reference state named "My Ref State 1", then an entry will appear in the reference states list displaying this name. It will also display the number of functions referenced and the number required if these values are the same for all referring systems. If they are not the same for all referring systems, then the word "varies" will appear instead of a number.

There are a number of actions that can be taken from the reference state list box. For example, right clicking on a displayed reference state presents the menu shown in Figure 2-8 below.

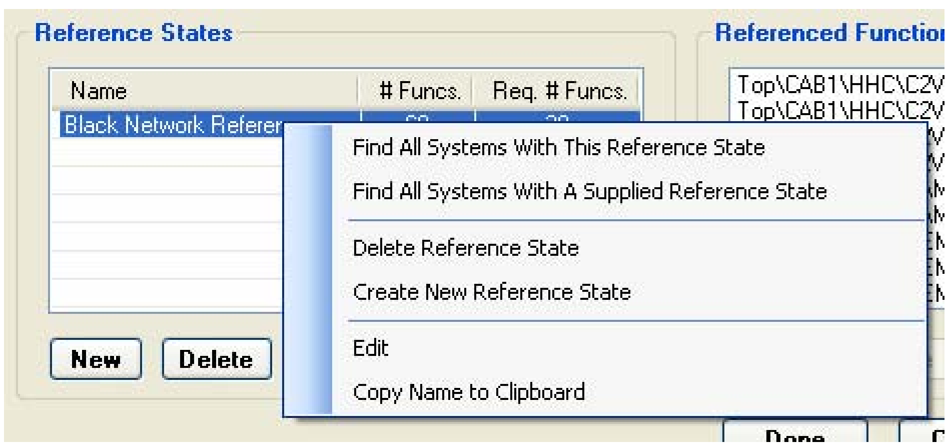

Figure 2-8. Reference State Item Popup Menu.

The *"Find"* options allow a user to perform a filtering operation and have the results presented for selection just as described above. Selections made in this way are applied to the referring systems tree. The *"Delete Reference State"* option allows a user to completely remove the selected reference state(s) from all checked referring systems. This has the exact same effect as the *"Delete"* button shown in the panel. The *"Create New Reference State"* option allows a user to define a new reference state to be duplicated into all checked referring systems. This has the exact same effect as the *"New"* button shown in the panel. The *"Edit"* option allows a user to modify the reference states name or required number of functions. Any modifications will be applied to all checked referring systems. Finally, the *"Copy Name to Clipboard"* option allows a user to copy the name of the reference state to the windows clipboard as text.

Right clicking on an empty line presents the menu shown in Figure 2-9 below. The menu items have the same meanings as above.

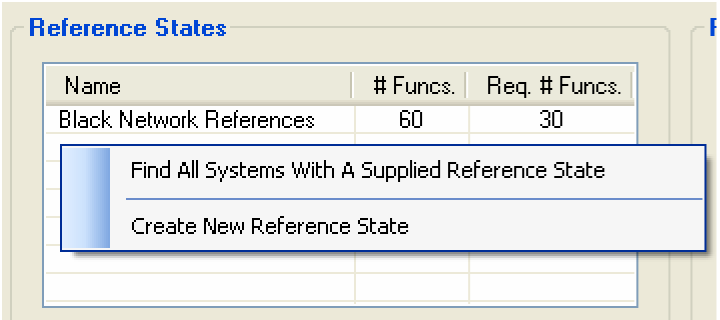

Figure 2-9. Reference State List Popup Menu.

Creating a new reference state either by using the popup menu option or by using the *"New"* button presents the user with a dialog in which the name for the new reference state and the number of operational function references for a true state are input. The dialog is shown in Figure 2-10 below.

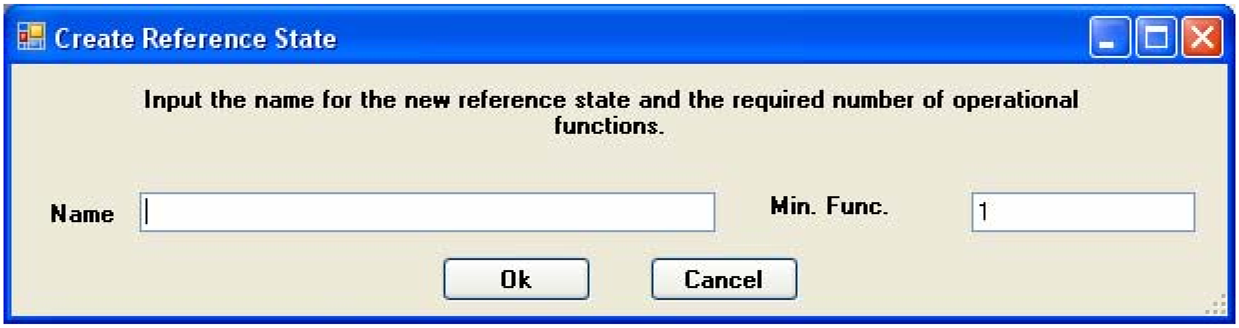

Figure 2-10. The New Reference State Creation Dialog.

Once this dialog is filled in and the *"Ok"* button is used, the new reference state will appear in the list and may be selected for the purpose of adding functions, etc.

The Referred-To System Section

This section houses the referred systems tree view. It is only sensible to select function area and function level nodes in this tree. They are the only node types to which reference states can refer. You will notice however that there are check boxes next to all nodes, not just the function area and level nodes. Selection of any of the parent nodes results in the selection of all child nodes. This feature makes it simple to select all functions in a particular branch of the force structure. Nodes checked in this tree can be added as referred to functions within a selected reference state.

There are operations that can be taken from within the referred systems tree view. For example, right clicking on a function node will bring up the menu shown in Figure 2-6 below.

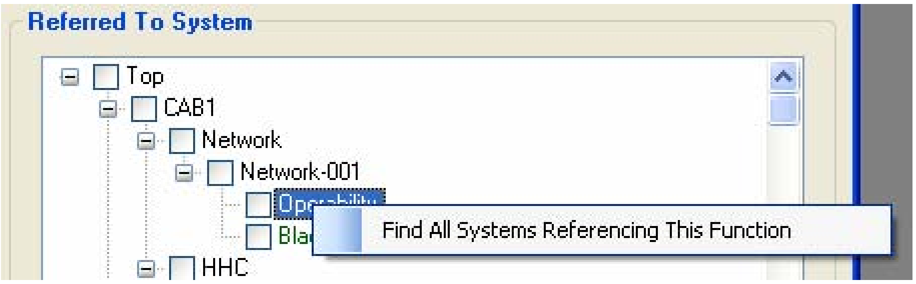

Figure 2-11. Referred System Tree Function Node Popup Menu.

The *"Find All Systems Referencing This Function"* option will present a list of all systems that have a reference state that references the chosen function. The user can then choose from the displayed systems and affect the selections **in the referring systems tree.**

Right clicking on a system node will bring up the menu shown in Figure 2-12 below.

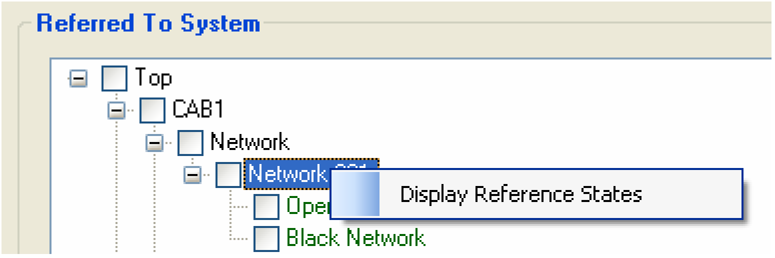

Figure 2-12. Referred System Tree System Node Popup Menu.

The "Display Reference States" option has the same effect that it has for the referring systems tree as shown in Figure 2-6.

The Referenced Functions Section

The referenced functions section is where the list of all functions referenced by a selected reference state are displayed. The reference states are selected in the Reference States section. As shown in Figure 2-1, this section consists of a list, two buttons, and a field. The list shows the full node paths to all referenced functions. The *"Add"* button allows a user to add all currently selected functions to the list of those referenced by the current reference state(s) and thus to the list of this section. The *"Delete"* button allows a user to remove functions chosen in the list of this section from the selected reference states and thus from this list, and the text field shows the current count of reference functions.

For any given reference state, there is a required number of operational functions in order for the reference state to be true. To help make this requirement possible, this form will recognize if there are too few referenced functions in a reference state for this to happen. If that is the case, the reference state will appear in red in the reference states section as shown in Figure 2-13 below. It is in this panel that a user can add the necessary functions to correct such a situation.

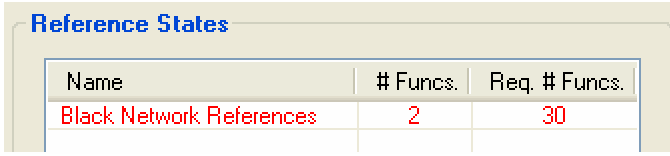

Figure 2-13. Incomplete Reference State Specification.

There are operations that can be taken from within the referenced functions section. For example, right clicking on a function path in the list will bring up the menu shown in Figure 2-14 below.

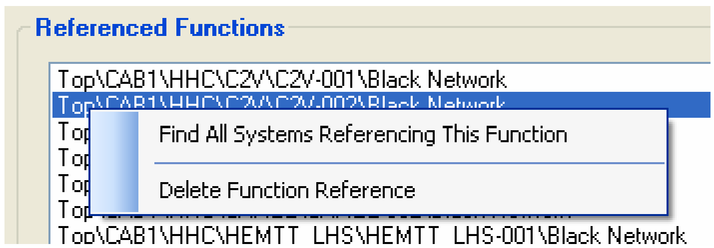

Figure 2-14. Referenced Functions List Pop Up Menu.

The *"Find All Systems Referencing This Function"* option has the same meaning as in the referred to systems section. The *"Delete Function Reference"* does the exact same thing as the *"Delete"* button of this section.

Creating Reference States

Now that each section of the form has been described, it is necessary to present the method by which a user will create and fill new reference states for a system. The steps that must be taken are:

- Choose the referring systems (see The Referring System Section).
- Create the new reference state (see The Reference States Section).
- Select the new reference state in the reference state list (see The Reference States Section).
- Choose the referred to functions (see The Referenced Functions Section).
- Add the chosen functions to the currently selected reference state (see The Referenced Functions Section).

Once these steps are completed, a reference state has been created and fully defined. Repeat this process for as many reference states and as many systems as desired. It is important to note that the resulting reference states are only elements that *may* be used in system functions but have not yet. The next section describes how to use these reference elements in equations.

2.3.5. System Function Definition

SoSAT has an existing capability for combining elements into system functions. However, as with the reference assignment, the existing capability had to be improved to support large scale use of reference elements. In particular, when creating and using many reference elements, there is a possibility that circular references can be created. A circular reference occurs when a function of a system refers (via a reference element) to function of another system which in turn refers back to the original function. When this is the case, the functions with these reference elements can never be resolved as each will wait for the other to update.

Consider Figure 2-15 below showing a simple example of this situation.

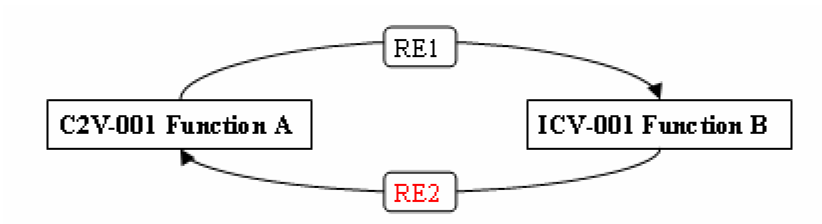

Figure 2-15. Simple Circular Reference.

Given that reference element RE1 is already part of C2V-001\Function A and that it references ICV-001\Function B, the addition of RE2 causes a circular reference. Circular references may be less obvious than the one above when for example there are multiple layers of references involved as shown in Figure 2-16 below.

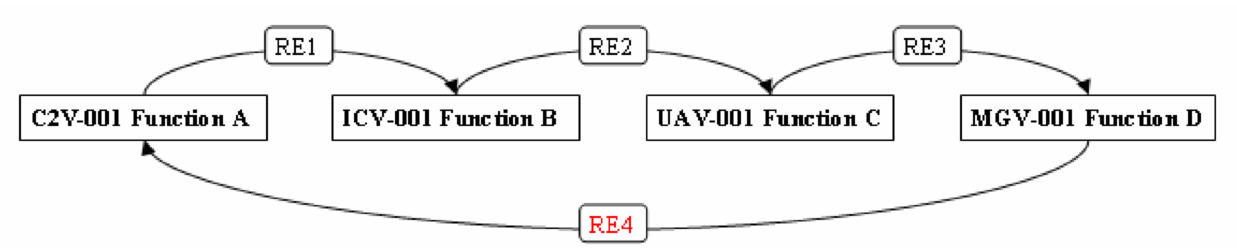

Figure 2-16. Multi-layer Circular Reference

Because a circular reference cannot be detected until a reference element is actually used in a function, the algorithm necessary to detect them had to be implemented in the form in which the functions are constructed.

Given a reference state for inclusion in a function equation, the algorithm operates by tracing through all reference elements of all referred functions of the supplied reference state until they have all been traced or a problem is found. So for example, consider the following scenario.

Given:

- A reference state called “Bad Reference State” for a system “C2V-001”.
- A reference state called “Black Network References” for a system “Network-001”.
- “C2V-001” has a function called “Black Network”.

- “Network-001” has a function called “Black Network”.
- “Bad Reference State” has “Network-001\Black Network” in its list of referenced functions.
- “Black Network References” has “C2V-001\Black Network” in its list of referenced functions.
- “Black Network References” has been added as a term to the equation for “Network-001\Black Network”.

An attempt to add the “Bad Reference State” element to “C2V-001\Black Network” function under these conditions would result in the following error dialog.

Figure 2-17. Example Circular Reference Detection Error Dialog.

What has happened is that the algorithm has detected that through the Bad Reference State, C2V-001\Black Network has referenced Network-001\Black Network which in turn references C2V-001\Black Network through the reference state Black Network References creating a circular reference. Figure 2-18 below shows this case in the form of Figure 2-15 above.

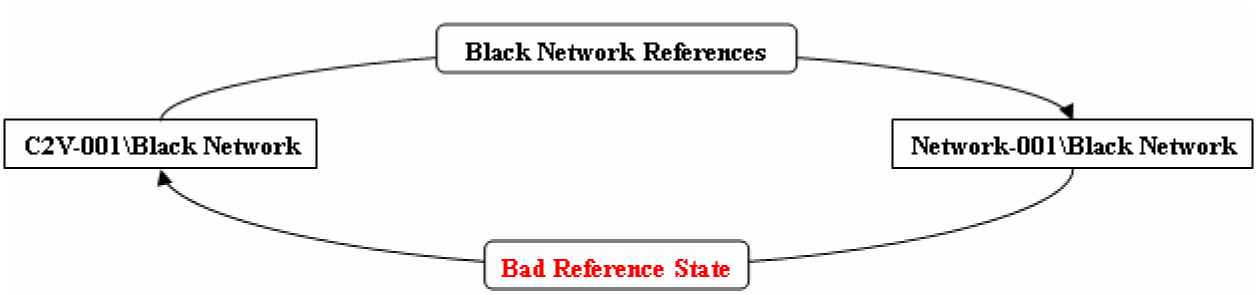

Figure 2-18. Circular Reference Path for Example Problem.

3. Example Problem

This chapter provides a detailed account of an example used to demonstrate the potential for the addition of network modeling to affect platform A_o.

3.1. Problem Description

The problem developed to demonstrate and test this new capability is based on the FCS BCT but is not a full scale representation of the BCT structure. In addition, the platforms that are represented are modeled using notional elements with reliability data designed to provide proper mean time between system abort (MTBSA) values for the platforms. The platforms from the standard BCT included in the simulation are shown in Table 3-1 below.

Table 3-1. Example Problem Platforms

Platforms	Description
C2V	Command and Control Vehicle.
CAMEL	Water Tank
FLEX	Storage Container
FRMV	Future Recovery & Maintenance Vehicle
HEMTT-LHS	Heavy Expanded Mobility Tactical Truck
HMMWV-SPT	High Mobility Multipurpose Wheeled Vehicle – Support
HMMWV-C2	High Mobility Multipurpose Wheeled Vehicle – Command and Control
ICV	Infantry Carrier Vehicle
M-1	
MULE-CM	Unmanned Ground Vehicle – Countermine
MV-E	Medical Vehicle - Evacuation
MV-T	Medical Vehicle – Treatment
PLS Trailer	Palletized Load System Trailer
Tank Rack	
UAV1	Type 1 Unmanned Aerial Vehicle
UAV2	Type 2 Unmanned Aerial Vehicle

The notional elements mentioned above can be thought of as representing the non-network related hardware of the platforms. In each case, there are 6 such elements labeled E1-E6. These elements provide the hardware failures that result in the need for spare parts and repair services. Each element type is a conglomerate representation of part failures as described in the list below.

- E1 – Failures of those parts that can be obtained from on-board inventories.
- E2 – Failures of those parts that can be obtained from the Forward Support Battalion (FSB).
- E3 – Failures of those parts that can be obtained from the Combined Arms Battalion (CAB).
- E4 – Failures of those parts that can be obtained from the Unit of Employment (UE).
- E5 – Failures of those parts that can be obtained from the Advanced Point of Debarkation (APOD)

- E6 – Failures of those parts that can be obtained from the Continental United States (CONUS)

A percentage of the total failures that may occur is assigned to each of these groupings. Based on the total MTBSA and these percentages, the MTBF of each element can be determined. The percentages have been supplied by the FCS program. Consider the example of a C2V whose total platform MTBSA is 432 hours which means its failure rate is 0.002315 failures per hour. Table 3-2 below demonstrates the calculation of the MTBF of the elements E1-E6 using the percentages supplied by FCS.

Table 3-2. Example Failure Rate Calculation for E1-E6 of C2V.

Element	Failure %	MTBF (= 0.002315 * Failure %)	Failure Rate (= $MTBF^{-1}$)
E1	1.0%	0.000023	43191.82
E2	24.0%	0.000556	1799.66
E3	5.0%	0.000116	8638.36
E4	18.0%	0.000417	2399.55
E5	26.0%	0.000602	1661.22
E6	26.0%	0.000602	1661.22

This process has been carried out for all platforms.

In order to obtain the parts and services, a platform must use its network function to contact suppliers. If the network function is not available for whatever reason, the orders cannot be placed and so the repairs cannot be made. Once the network function is again available, repairs can be made. These delays that can be caused by network outages are the primary means by which the network affects A_o in this problem.

In addition to the notional platform hardware elements, there are elements representing the hardware used for communication. The hardware components used for communication are the parts of the JTRS radios. Each platform is given a JTRS radio and it is used to send information via radio frequency (RF) to other platforms. The components used to represent the JTRS radio in this problem are shown in the list below.

- Universal Transceiver #1
- Universal Transceiver #2
- Universal Transceiver #3
- Universal Transceiver #4
- Mount
- Local Control Display Device
- Ground Vehicle Adapter
- Networking INFOSEC Unit
- Wide Band Power Amplifier #1
- Wide Band Power Amplifier #2
- Wide Band Power Amplifier #3
- Wide Band Power Amplifier #4
- VHF/UHF Power Amplifier
- Antenna #1
- Antenna #2
- Cables

All platforms have one of each of the components listed with the exception of the Ground Vehicle Adapter. Only ground vehicles have that component.

3.2. Simulation Input

This section contains actual screen shots of the SoSAT SMOSimulation tool. The input of the problem into the tool begins with the setting of the simulation parameters. This includes the total length of the scenario, the time step, the random seed, etc. as shown in Figure 3-1 below.

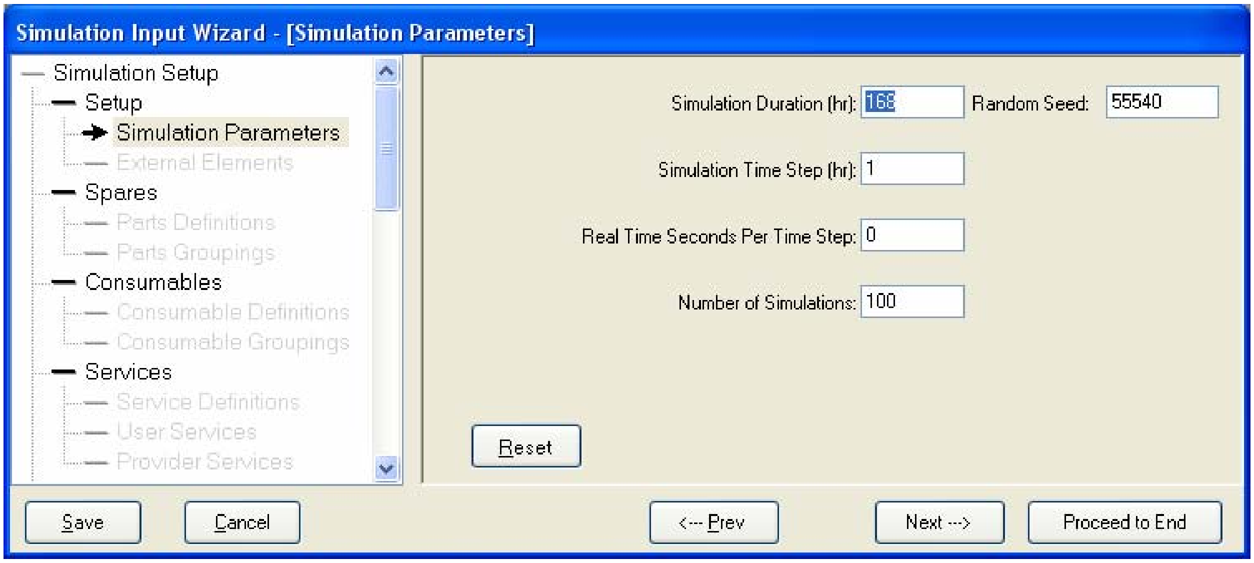

Figure 3-1. Input of Simulation Parameters for the Network Example Problem.

As can be seen from this figure, the total simulation time will be 168 hours (7 days) and the final results will be the aggregation of the results of 200 individual simulation replications. Replicating the simulation multiple times provides statistical significance to the results.

The next input step relevant to this problem is the definition of the parts used. This includes all the parts required by our notional elements and those required by our JTRS radios. Figure 3-2 below shows a small subset of the total collection of parts used in this problem but the sampling is sufficient to present the necessary information.

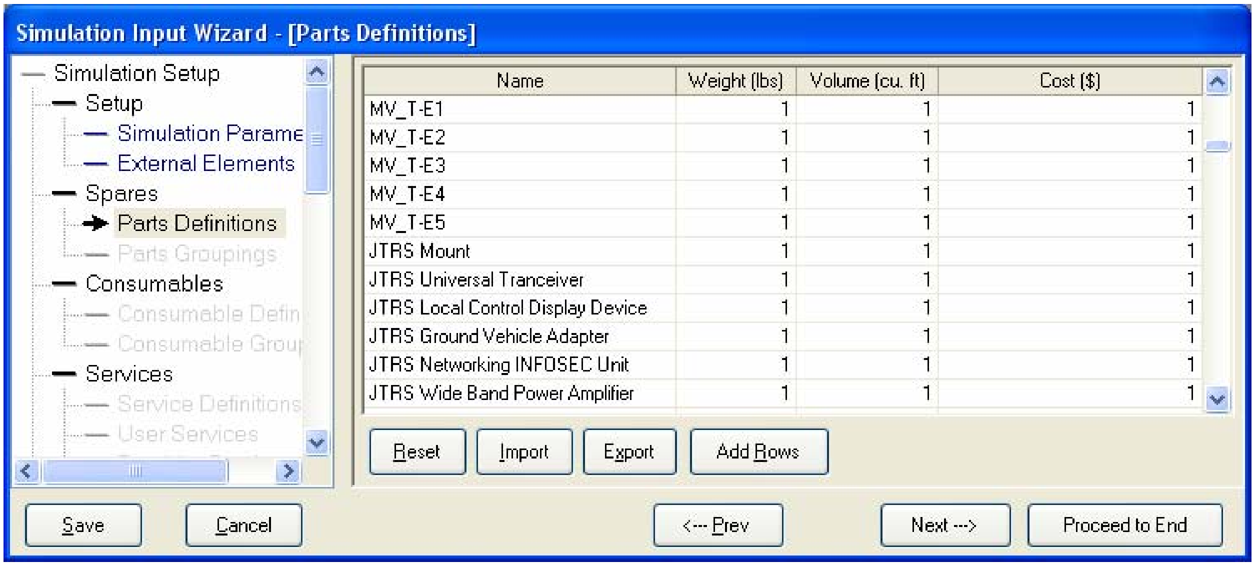

Figure 3-2. Input of the Part Definitions for the Network Example Problem.

The figure shows the parts for the notional elements for any MV-T instances and parts for some of the JTRS hardware elements for any of the network enabled platforms. The values for the weight, volume, and cost are set to unity since for the purposes of this simulation they are unnecessary.

The next required inputs are the various inventories that the different platforms may hold. These inventories contain the spare parts that may be used by the platforms themselves (self-supply) or provided to other platforms that need them.

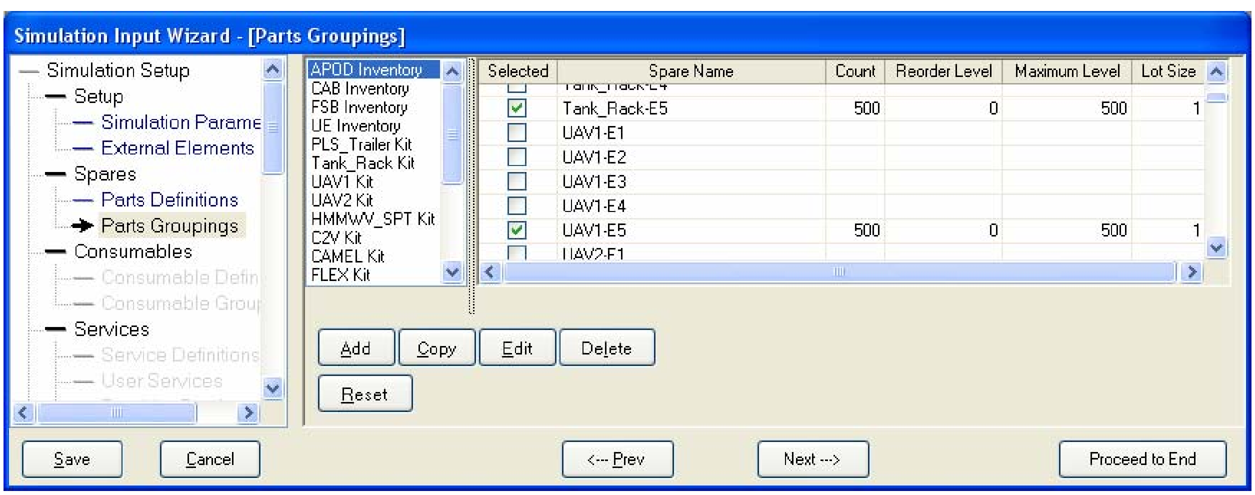

Figure 3-3. Input of the Part Groupings for the Network Example Problem.

It is important to have each part that may be required by any of the platforms represented in at least one inventory. The figure shows that the APOD inventory for example contains 500 spare Tank_Rack-E5 parts and 500 UAV1-E5 parts among other things.

Once the parts and inventories are defined, the next step is to define all the consumables that exist in the simulation. Consumables are any commodities that are used up by platforms and must be periodically re-supplied. Examples can be seen in Figure 3-4 below and include things like ammunition, fuel, water, etc.

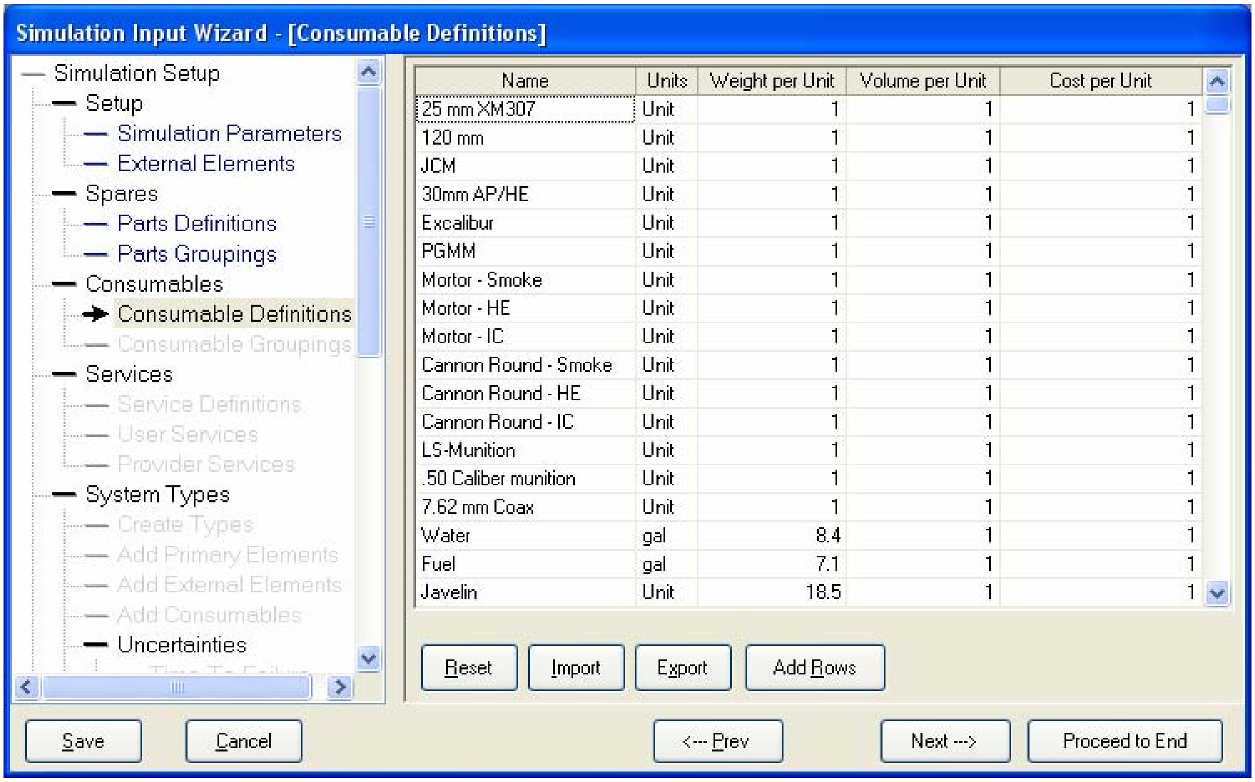

Figure 3-4. Input of the Consumable Definitions for the Network Example Problem.

The resupply of consumables, like the obtaining of spare parts, is something that will require communication between platforms and thus will require network functionality. As with the parts, the properties of the consumables such as weight per unit, cost per unit, etc. are unimportant to this simulation and most have been given a nominal value of unity.

As with parts, consumables are supplied by platforms which have an inventory. The next input step is to define those inventories. Figure 3-5 below shows that input for this problem.

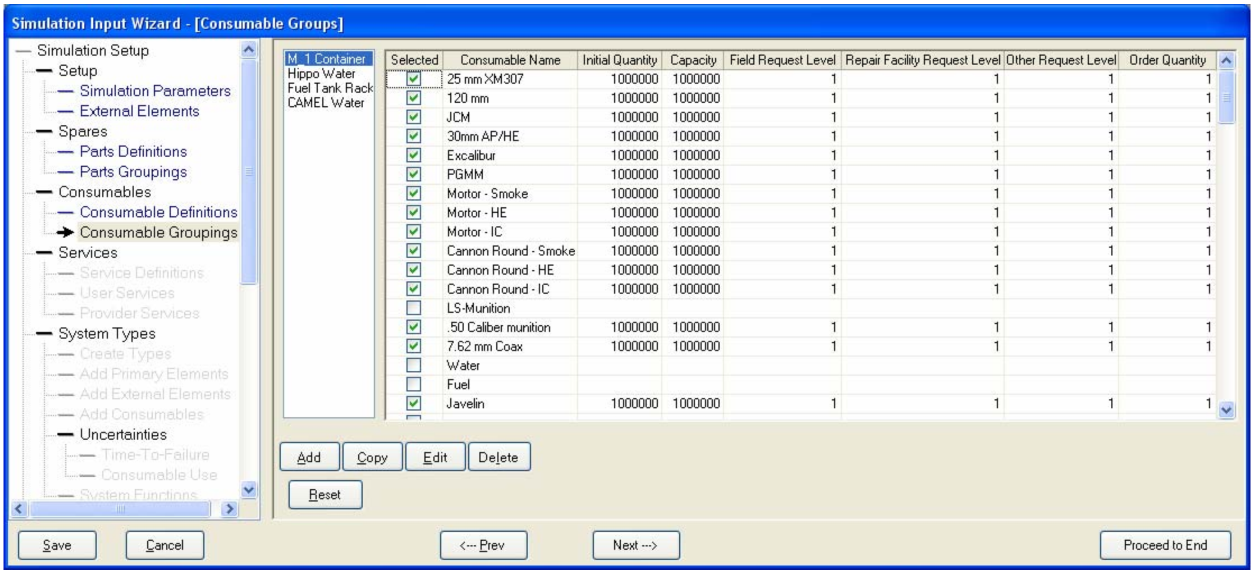

Figure 3-5. Input of the Consumable Groupings for the Network Example Problem.

Once the parts and consumables are defined, the next step is to define the services that are required within the force structure. These can be any sort of service. Examples might include whatever services are needed to replace failed components, towing, etc. Figure 3-6 below shows the input of the services used in this example. The cost values are unimportant as they are not used in this problem.

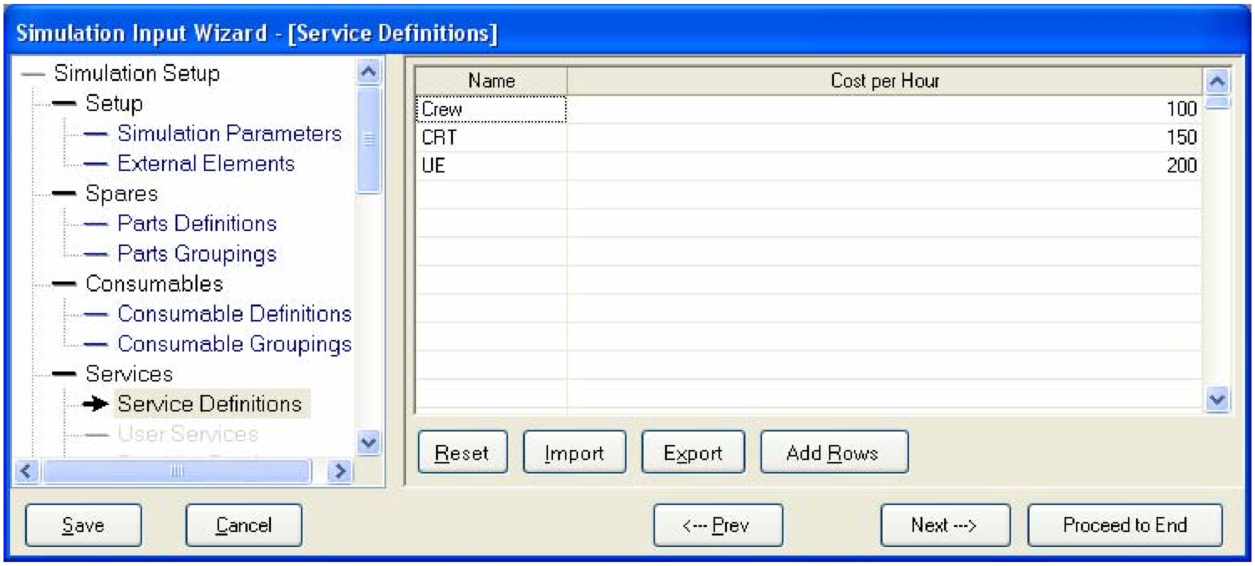

Figure 3-6. Input of the Service Definitions for the Network Example Problem.

As with the parts and consumables, it is necessary to create groupings of services that can be assigned to providers. That is the meaning of the input shown in Figure 3-7 below.

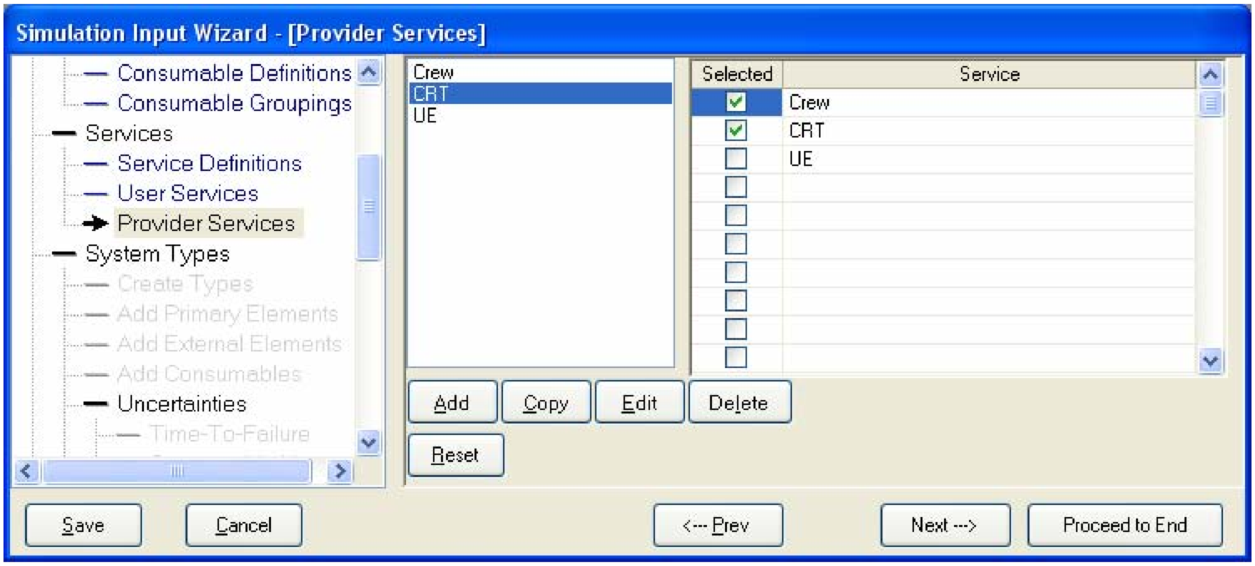

Figure 3-7. Input of the Provider Services for the Network Example.

The figure shows that any platform that is assigned the CRT service group will be able to provide both Crew and CRT services.

It is now necessary to define the actual system types that make up the force structure. These are the actual platforms that take part in the simulation. Figure 3-8 below shows the system types for this example problem.

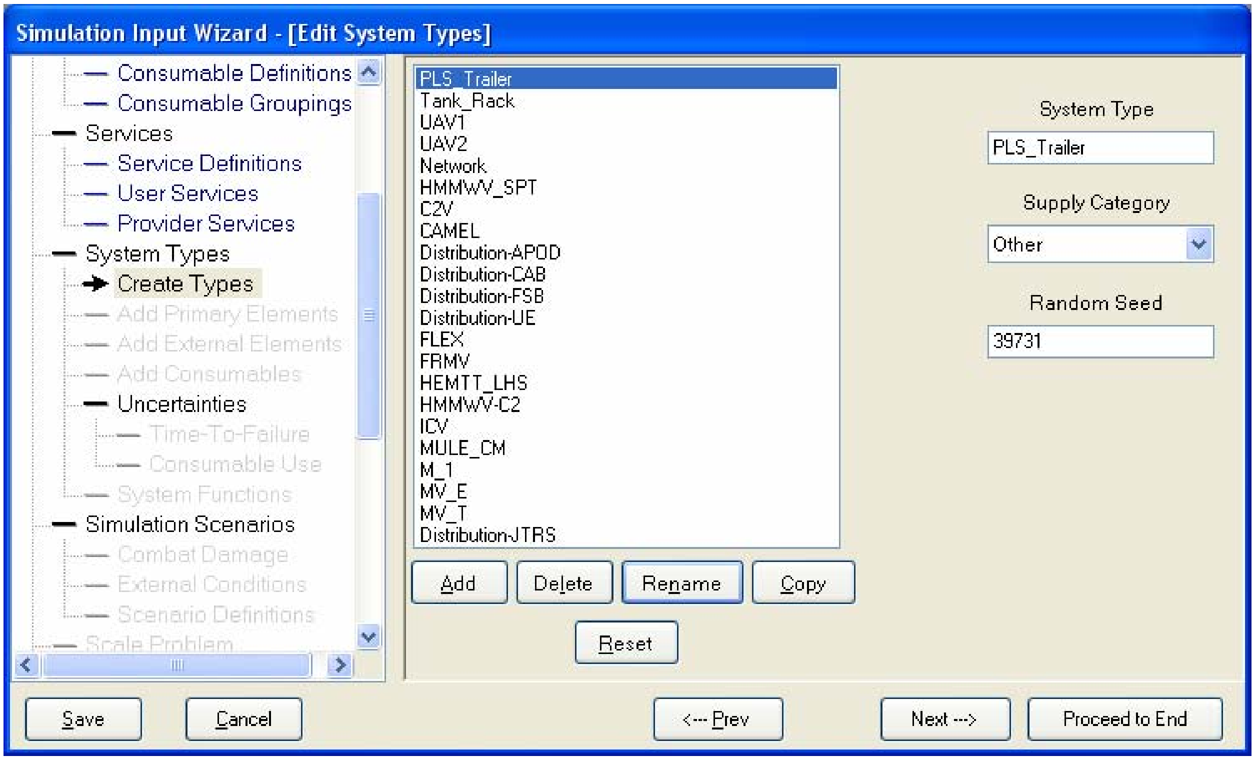

Figure 3-8. Input of the System Types for the Network Example.

The input shown in the above figure is little more than the naming the various systems. In the following steps, the platforms will be filled in with the data necessary to make them behave in desired ways. To that end, the next input step is to define the primary elements that will make up the platform functions as described in Section 2.1.1.

Figure 3-9 below shows a subset of the elements used to define the functions of C2Vs. Many inputs are required for each element including where and when it ages, where and when can be repaired, the failure and repair distributions, etc. The figure below shows the "Time-to-Fail Distribution" and associated failure rate distribution parameter for some of the parts used in the C2Vs.

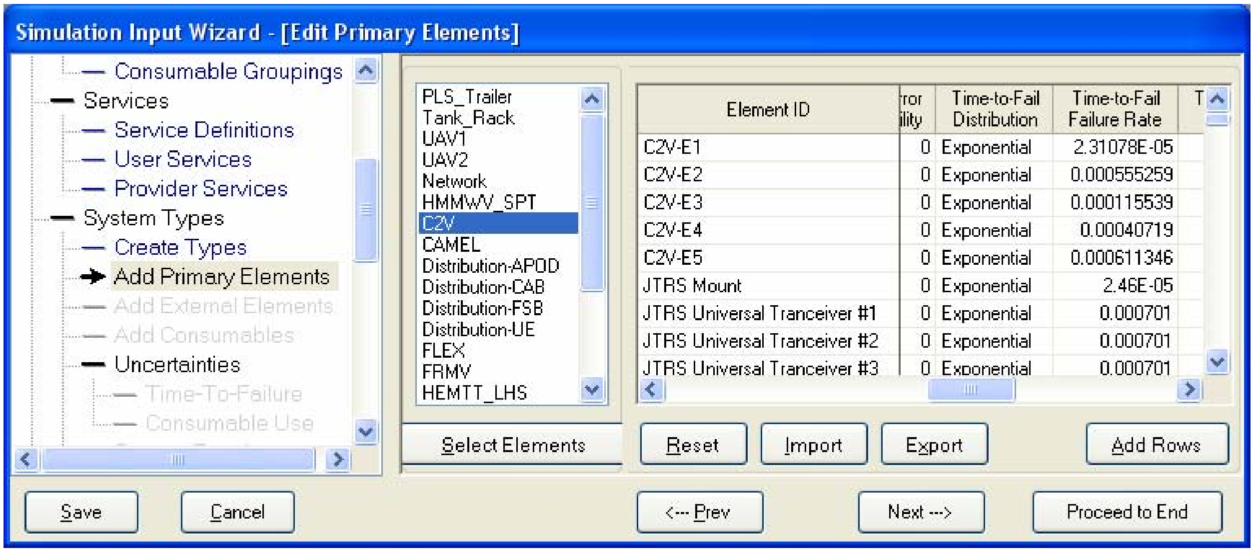

Figure 3-9. Input of the Primary Elements for the Network Example.

Of particular interest to this problem is the fact that the JTRS radio components do not show the need for spare parts or services as shown in Figure 3-10 below.

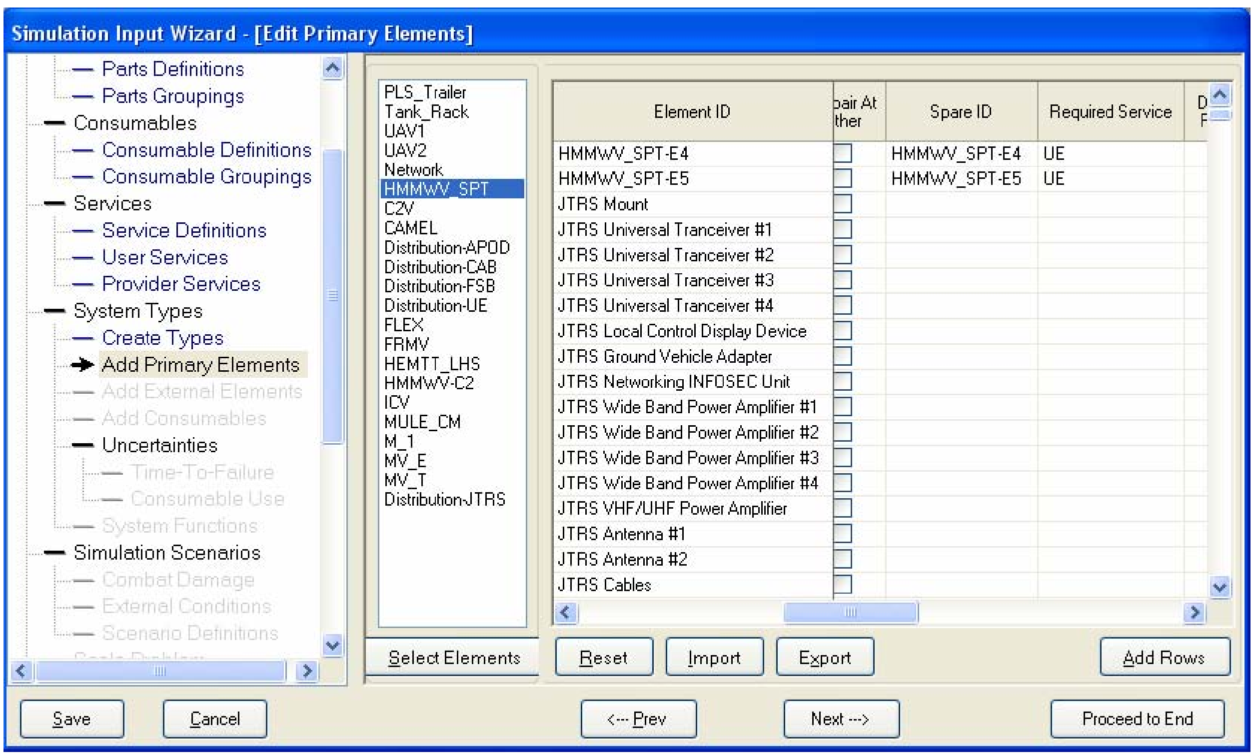

Figure 3-10. Input of the Primary Elements Without Spares for the Network Example.

The reason for this is to prevent a conundrum whereby the network is required to order a part or service to fix the hardware that is required for the network to be available. The justification for this is that in a real battle situation, there are alternative means of relaying order information in an emergency such as by physically contacting a nearby unit.

The next step is to assign consumables to platform types. Figure 3-11 below shows that, among other things, the C2Vs require fuel and water. The input includes information such as how much they can hold, how much they order if they run low in various locations, when the consumable is used, etc.

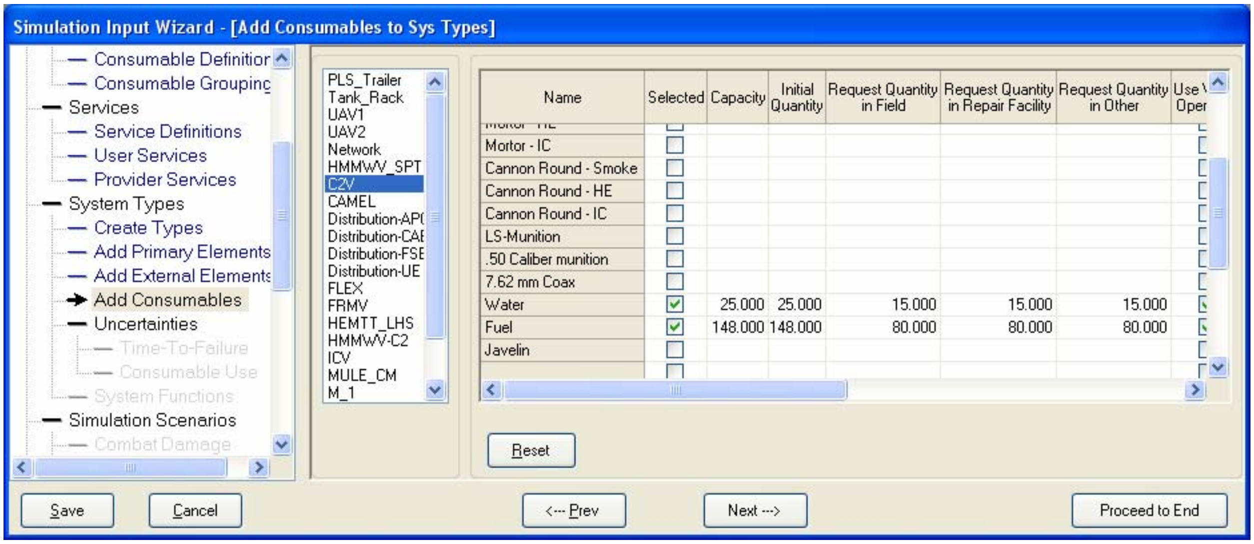

Figure 3-11. Input of Consumables Required by Platform Types in the Network Example.

Now that all the parts and consumables have been defined and prepared for use as elements, it is time to define the platform functions. SoSAT requires that all systems have a function called "Operability". In addition to the Operability function, all platforms are given a function called "Black Network" for this example. This function is meant to represent the platform's ability to communicate with other platforms via the network. The form of the Black Network function is important to this example and so it will be described in detail. Figure 3-12 below shows the equation for a C2V's Black Network function.

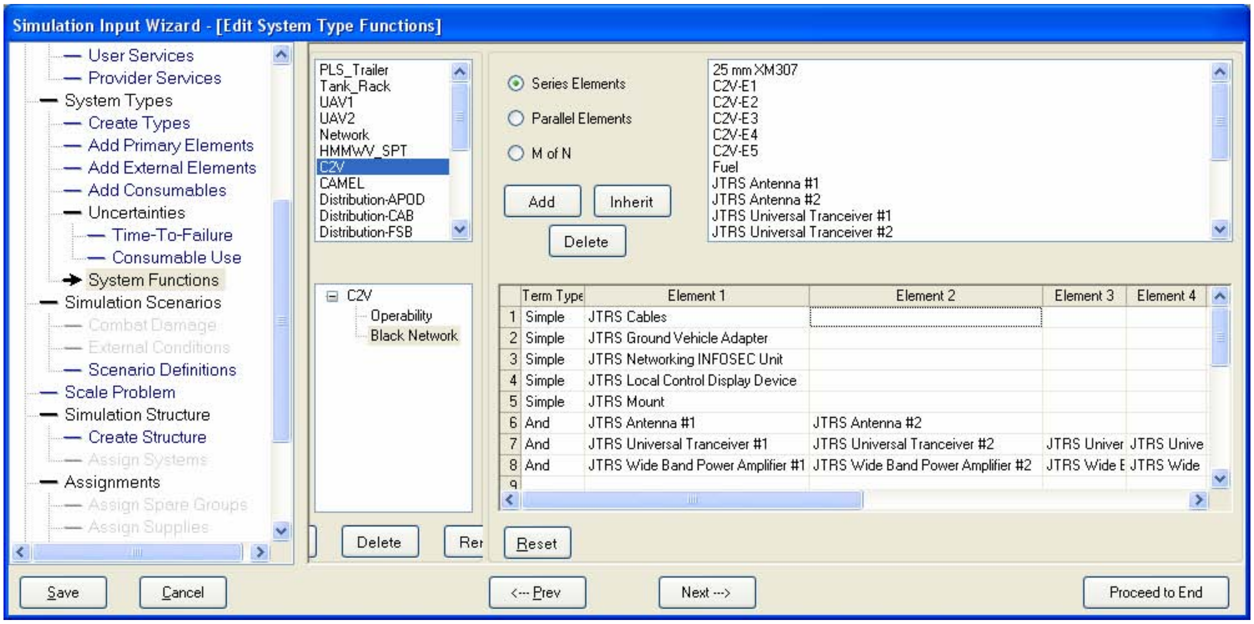

Figure 3-12. Input of the Platform Functions of the Network Example.

From the figure, it can be seen that the black network function has both simple (serial) terms and compound (parallel) terms. The function can be read as:

$$\text{Black Network} = \text{Cables} \,\&\, \text{Ground Vehicle Adapter} \,\&\, \ldots \,\&\, (\text{Antenna \#1} \,|\, \text{Antenna \#2}) \,\&\, \ldots \qquad \textbf{(3-1)}$$

where & represents the AND operator and | represents the OR operator as they did in Section 2.2. The Black Network function of all platforms is the same with the exception of the use of a Ground Vehicle Adapter. Only ground vehicles have that element.

The next step is to input the various operation scenarios that the platforms will use. These define where the platforms will be for each hour of the total mission and what their desired state of operability is during those periods of time. Figure 3-13 shows the MGV Scenario used by all ground vehicles in this example.

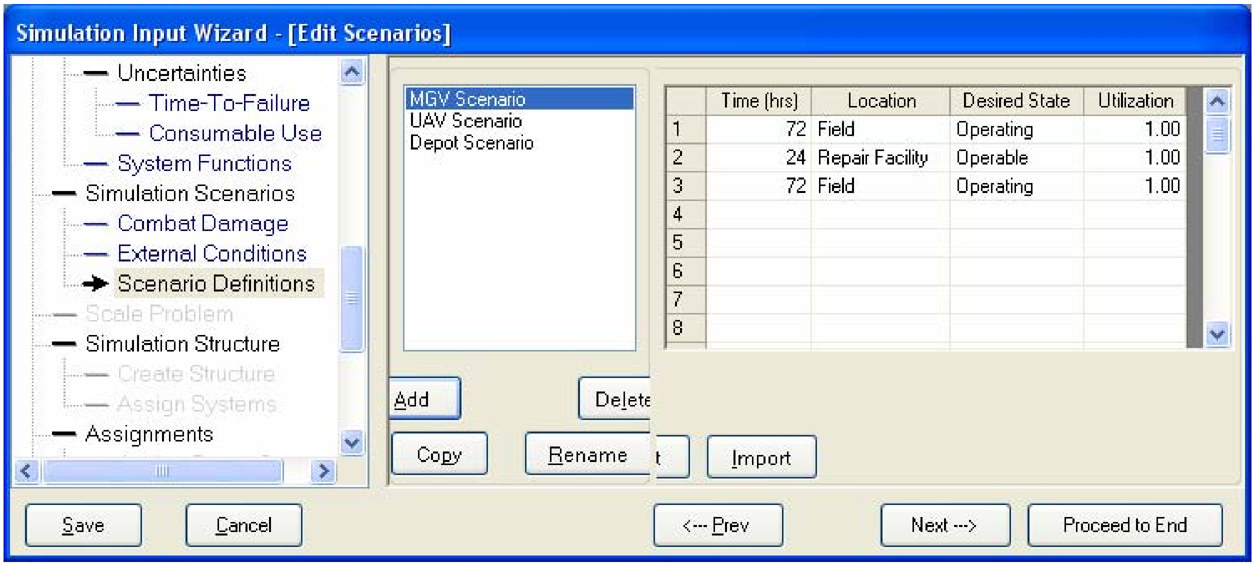

Figure 3-13. Input of the Scenario Definitions of the Network Example.

The next input step is to specify the number of instances of each platform type that will take part in the simulation. As an example, Figure 3-14 shows that there will be 4 PLS Trailers, 2 Tank Racks, etc. Once all quantities are specified, the *Scale* button will cause the creation of the actual instances.

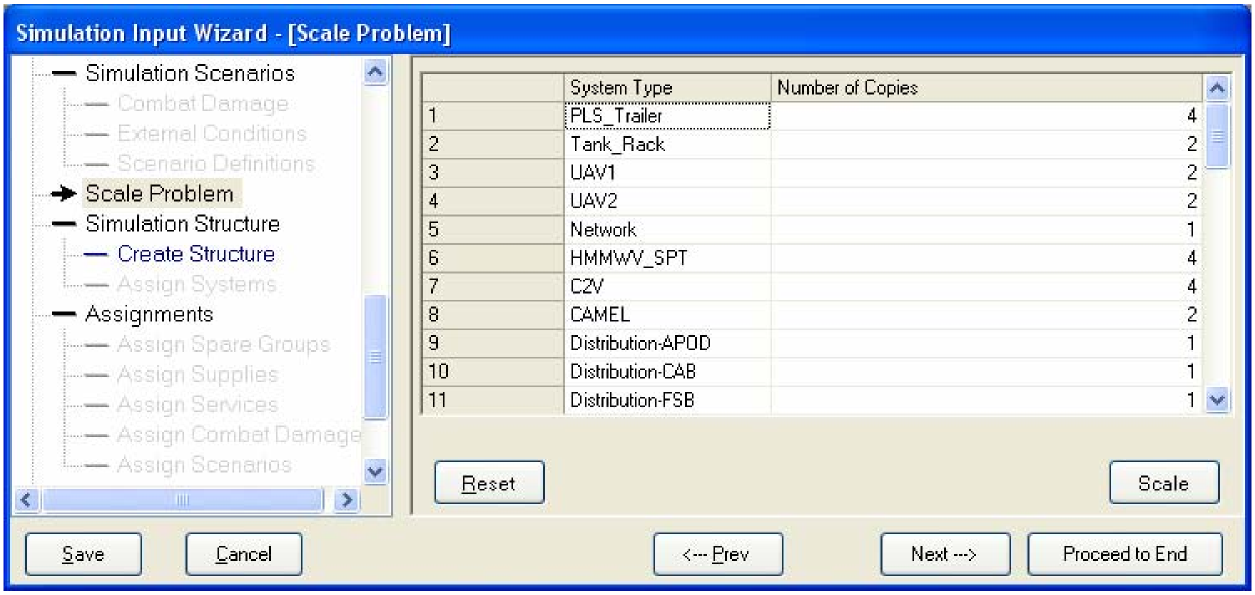

Figure 3-14. Input of the Platform Instance Counts of the Network Example.

It is now necessary to define the force structure for the example. This is the organization of the platforms into the command hierarchy. Although not required, it is good practice to create an echelon for each platform type as can be seen in Figure 3-15 below.

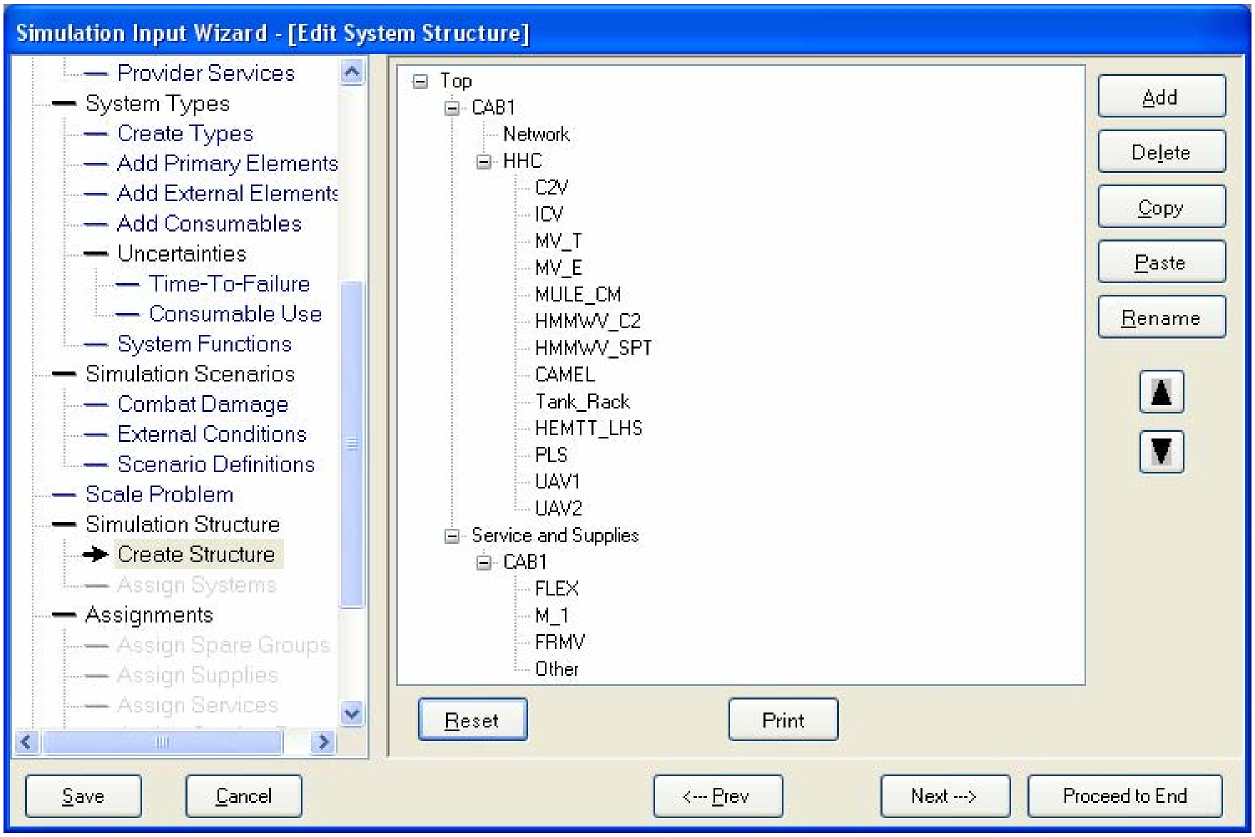

Figure 3-15. Input of the Force Structure of the Network Example.

Now that the various echelons are defined in the force structure, it is time to assign the platforms to them. In this example, all instances of the platforms will be assigned to the echelon of their platform type with the exception of the distribution depots. They will all be assigned to the "Other" echelon. Figure 3-16 below shows some of these assignments.

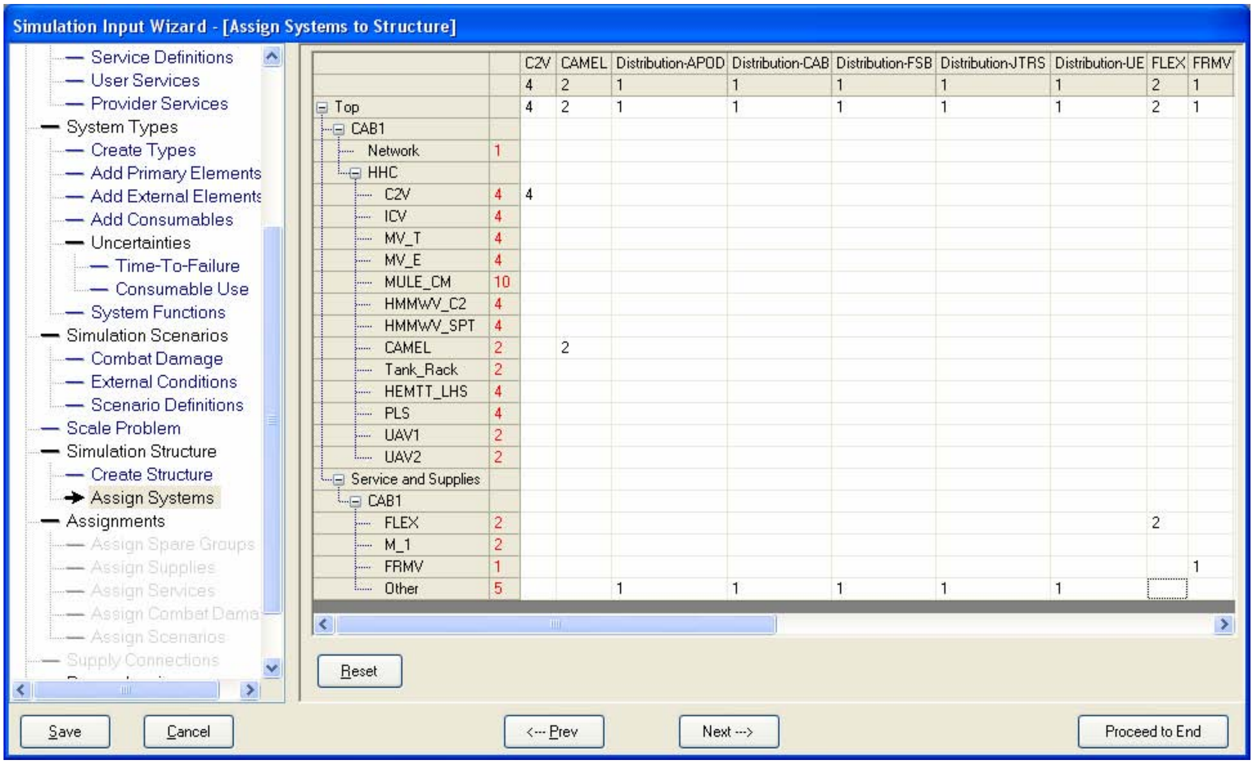

Figure 3-16. Input of the System Assignments of the Network Example.

Figure 3-17 shows the assignment of spare part groups to each of the platforms. This is the next step required for the input of the example problem.

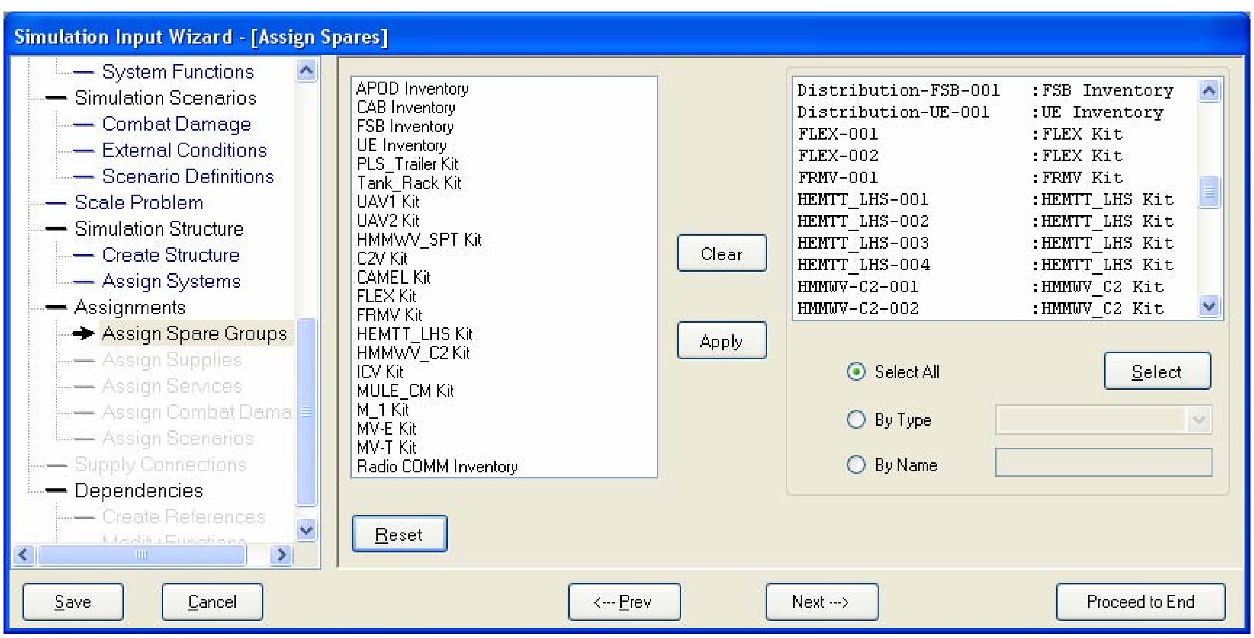

Figure 3-17. Input of the Spare Inventory Assignments of the Network Example.

Likewise, Figure 3-18 shows the assignment of consumable inventory groups to various platforms.

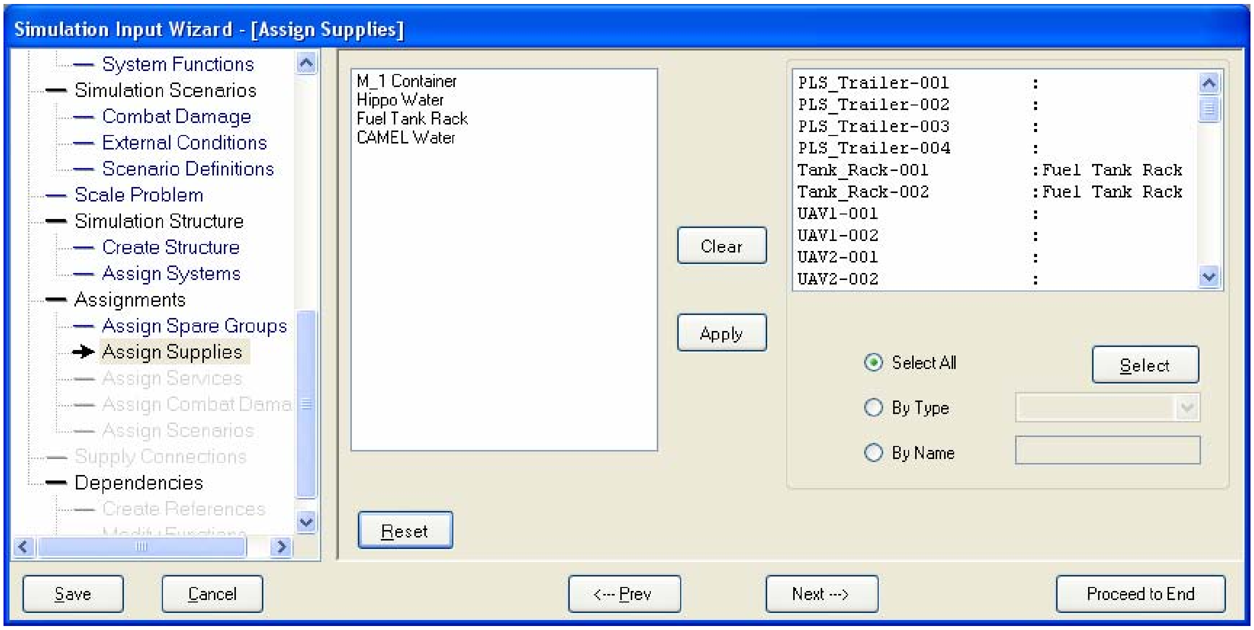

Figure 3-18. Input of the Consumable Inventory Assignments of the Network Example.

Figure 3-19 shows the assignment of service types to the platforms that are capable of providing them.

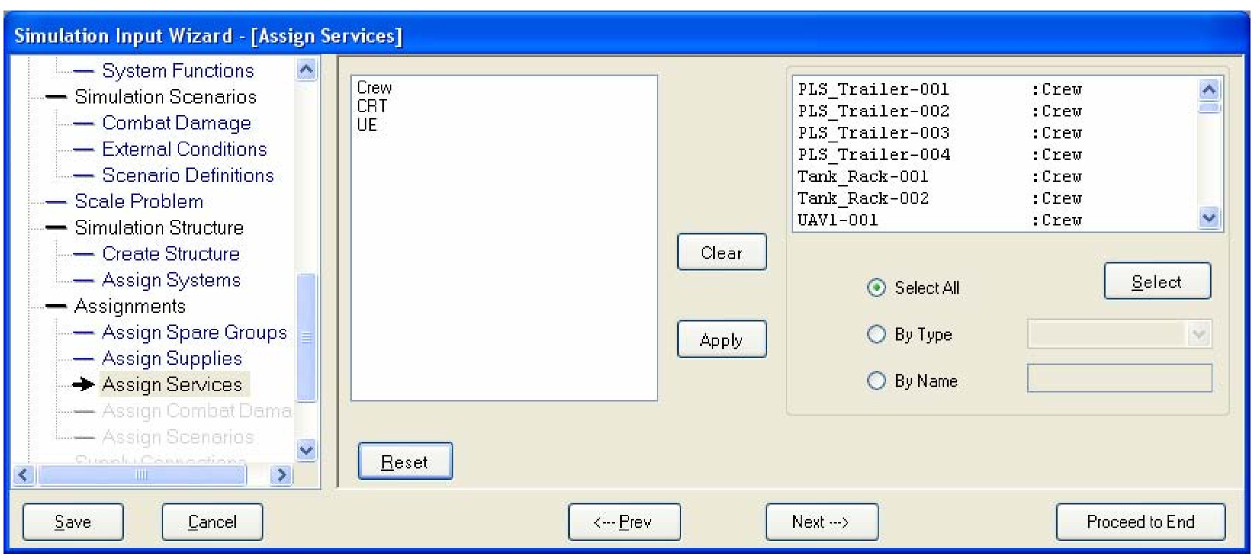

Figure 3-19. Input of the Service Assignments of the Network Example.

The next step is to assign a scenario to each platform. Figure 3-20 shows for example that the PLS Trailers are assigned the Depot Scenario which in this example means that they will be in the field and should be operational for the entire 168 hours of the operation.

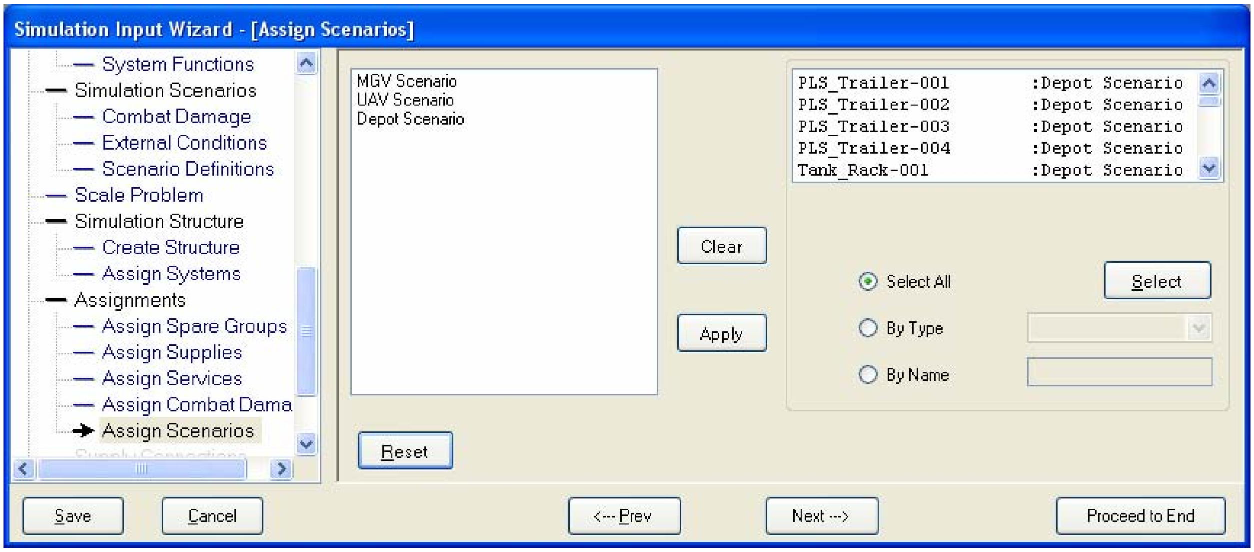

Figure 3-20. Input of the Scenario Assignments of the Network Example.

Now that each platform has been assigned the parts, consumables, and services that it can supply, it is necessary to specify which platforms can supply other platforms with those commodities and which can self-supply. Figure 3-21 below shows the form used to do this.

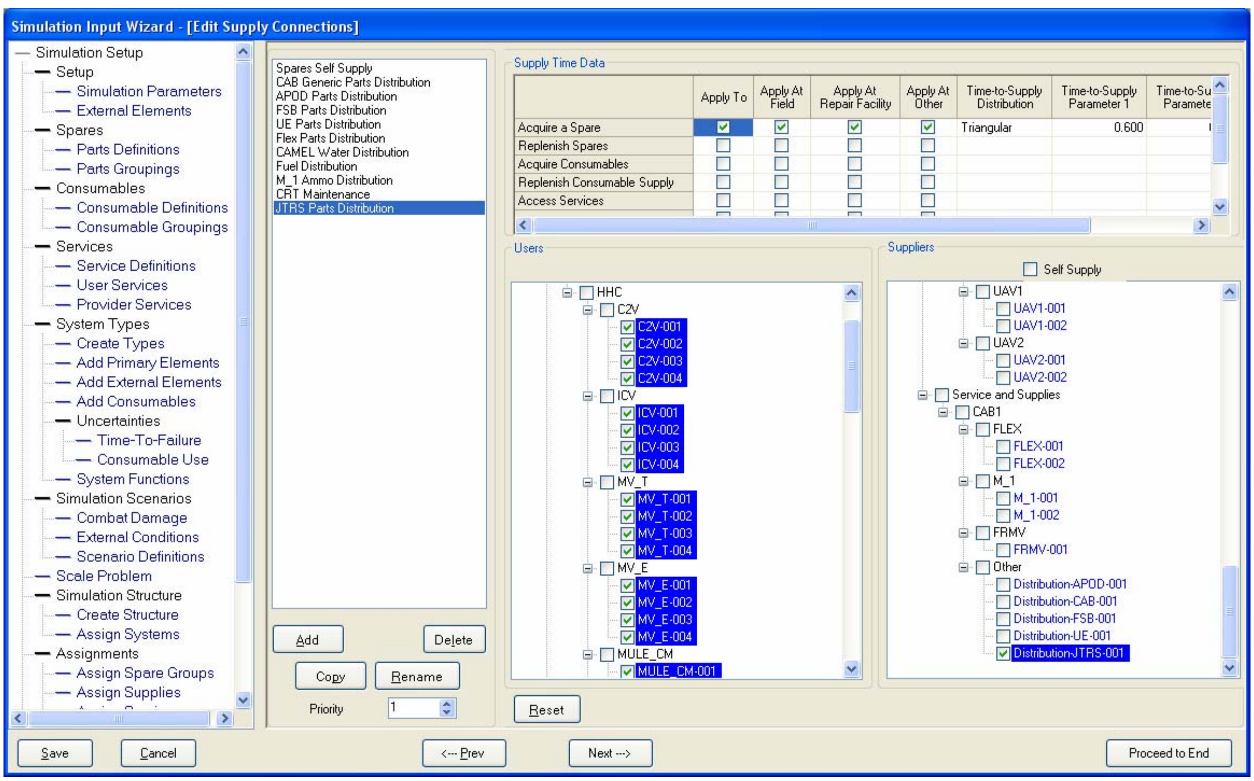

Figure 3-21. Input of the Supply Connections of the Network Example.

The figure shows that all JTRS radio components are supplied by the Distribution-JTRS depot system.

Now that all the functions for each of the platform instances have been defined, it is possible to create reference elements (see Section 2.1.1). This is very important to this example because it is how the functions of the Network system type are defined. Figure 3-22 below shows that the Network-001 system has a reference state that refers to 60 individual platform Black Network functions and requires 58 of them to be operational in order for the reference element to be in a true state.

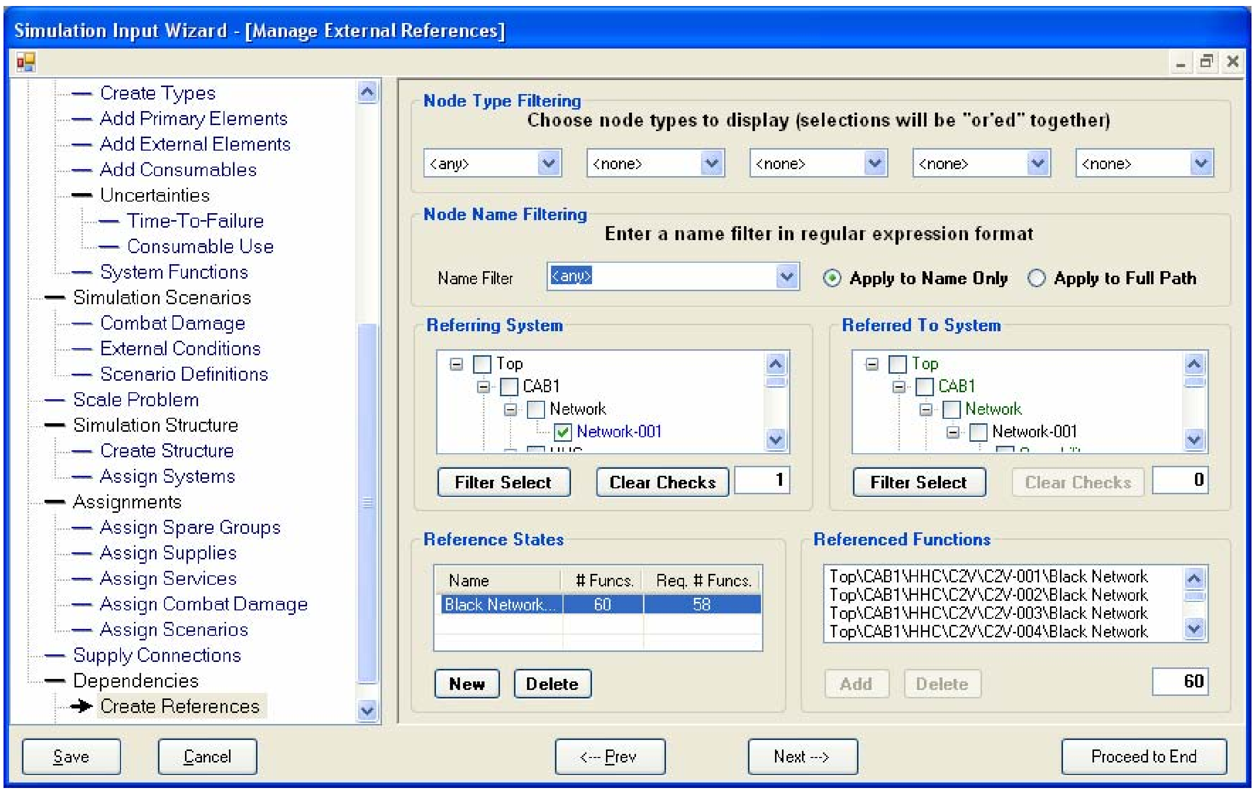

Figure 3-22. Input of the External References of the Network Example.

The requirement of 58 is likely an unrealistically high value but given the high degree of reliability of the components and the relatively short duration of the mission, it is necessary in order that this reference element have any impact on the A_0 of the Network-001 system in this example.

The final step in the input process is to incorporate these new reference elements into the system functions. The only system functions that have to be altered are those of any Network platforms. In this example, we only have the one Network platform, Network-001. Figure 3-23 below shows how the Operability function of the Network-001 system has been altered to contain the "Black Network References" reference element created above as the sole element. The Black Network function is identical in this example.

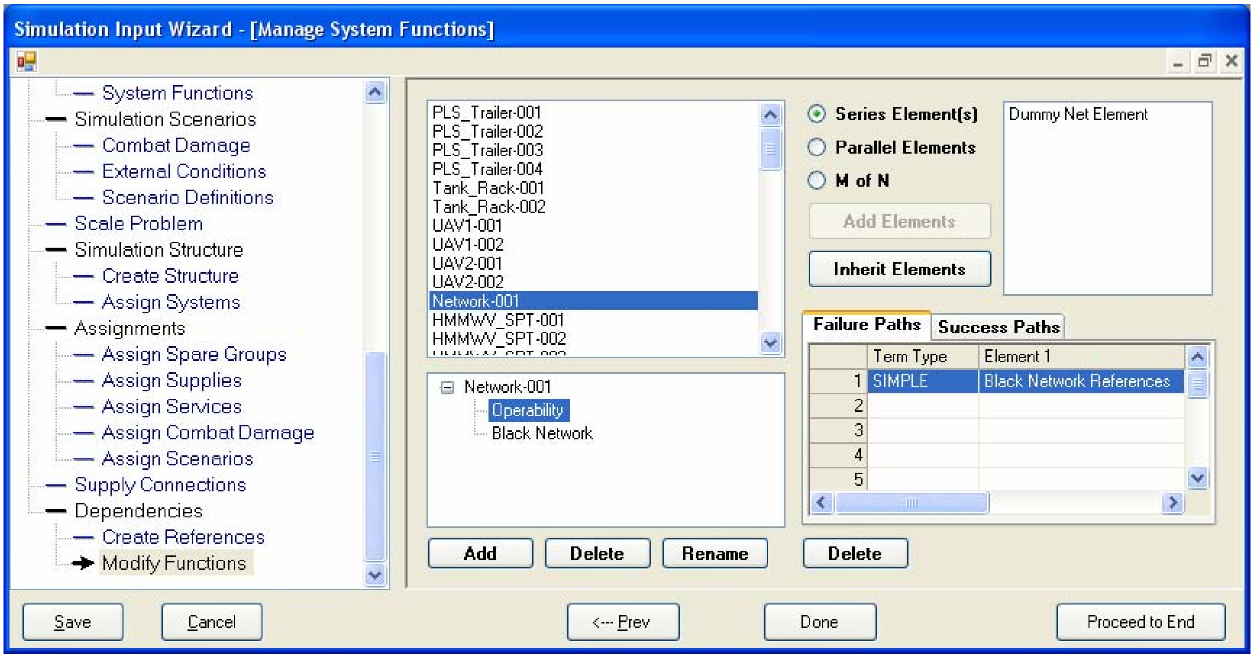

Figure 3-23. Input of the System Function Modifications of the Network Example.

Now that the input is complete, the next step is to run the simulation and gather the results. The next section details the results found and the effect of incorporating network hardware reliability failures on A_0 for the various platforms.

3.3. Results

In order to determine the effect of the Network on our systems, comparisons were made between the results of running the simulation with the network requirements enabled and running it with them disabled. This was done for various simulation settings to help qualify the results. While various data are manipulated to see the effects of the network, for all simulations, a mission consists of a 72 hour combat phase, a 24 hour replenishment phase, and another 72 hour combat phase totaling 168 hours. To give statistical significance to the results, each run of the simulation consists of some number of replications of the aforementioned mission. The results of each replication are combined to create the final statistics.

The first set of results was generated using the FCS supplied failure data for both the network and non-network platform hardware. The A_o results are the average of 5 runs of the simulation each of which performs 200 replications.

Table 3-3 below shows the results of those runs. Each row of the table represents the “roll-up” for all platform instances of the indicated type.

Table 3-3. Observed Effects of the Network with Notional but Representative Data.

Platforms	Operational Availability		Difference (%)
	Without Network Effects	With Network Effects	
Top	76.24%	76.07%	-0.23%
C2V	78.62%	78.16%	-0.59%
CAMEL	86.76%	86.90%	0.16%
FLEX	100.00%	100.00%	0.00%
FRMV	65.22%	65.80%	0.89%
HEMTT-LHS	77.84%	75.84%	-2.57%
HMMWV-SPT	69.02%	69.36%	0.49%
HMMWV-C2	67.14%	66.04%	-1.64%
ICV	75.40%	77.12%	2.28%
M-1	100.00%	100.00%	0.00%
MULE-CM	87.96%	87.50%	-0.52%
MV-E	39.38%	39.66%	0.71%
MV-T	41.60%	42.48%	2.12%
PLS-Trailer	99.96%	99.76%	-0.20%
Tank-Rack	96.44%	96.92%	0.50%
UAV1	73.24%	72.42%	-1.12%
UAV2	45.62%	45.56%	-0.13%

The difference column shows the **percent** difference between A_o without network effects and A_o with network effects, not the absolute difference. The calculation is performed as:

$$\frac{[A_o\ with]-[A_o\ without]}{[A_o\ without]} \qquad \textbf{3-2}$$

This is stated to prevent any confusion that may be caused for persons knowing that A_o itself is given as a percentage. Therefore, one could legitimately report the absolute difference in A_o as a percentage as well. That is not what is reported here.

The results show that network hardware reliability has virtually no impact on the overall A_o of the notional BCT for a mission as short as the one used here. Consider the **Top** rollup difference value of -0.23%. This means that the A_o decreased by this amount when taking network hardware failure into account. While this result is sensible, consider that 0.23% difference corresponds to an absolute difference of 0.17% (0.0017) in A_o in this case. Considering that the average standard error in the means used to compute these average A_o's is 0.0042, it is clear that a delta of 0.0017 is well within the noise of the simulation and therefore, the network failures had no overall effect.

Given the high level of reliability of the platforms as well as the network functions, this is as expected. In order for the network to have an impact, network failures must overlap other failures that not only affect A_o but require ordering of parts, consumables, etc. An important note is that all components in this notional BCT use exponential distributions to simulate probabilistic failures. Therefore, all failures are "timeless" and may occur at any point during a simulation replication with equal likelihood. This is important because it means that failures of the components are not biased toward any particular segment of the timeline. If that were the case, then it would be a factor when attempting to determine the probability of overlap of network and other hardware failures.

Also note that an overlap must be such that the network is needed at some point or it has no effect. This most often requires that the network hardware must fail prior to the failure of the non-network hardware and remain failed through at least part of the hardware failure as shown in Figure 3-24 and Figure 3-25 below.

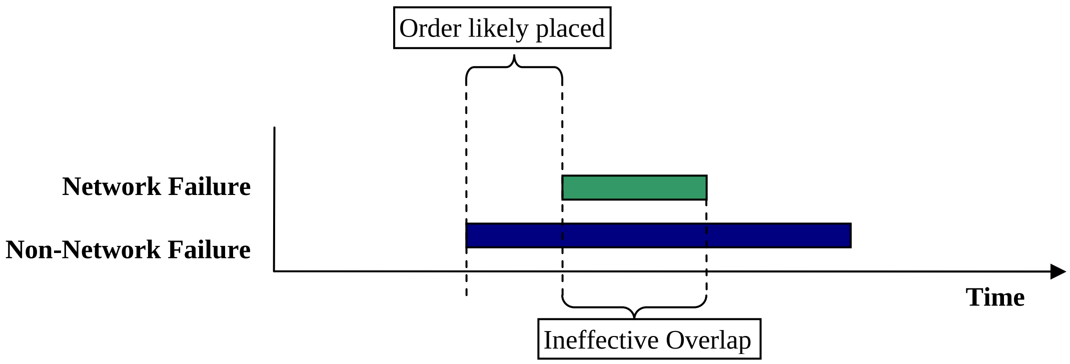

Figure 3-24. Demonstration of *ineffective* component failure overlap

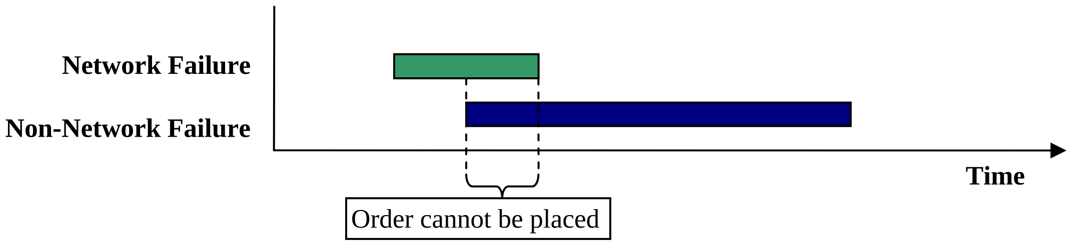

Figure 3-25. Demonstration of *effective* component failure overlap

This is not strictly true since it is possible that for other reasons such as depleted inventories or inoperable suppliers, an order may not get placed in the portion of Figure 3-24 labeled "Order likely placed". An attempt to place the order again may then occur in the time span labeled "Ineffective Overlap" and fail due to lack of network availability.

Consider the example of the C2V platform type. The results of the SoSAT runs *without* network effects show that for any given simulation replication, there is an approximately 72% chance that some C2V will experience a non-network hardware failure. Therefore, in this notional BCT where there are 4 C2V instances, slightly less than 3 C2V failures occur on average during a replication. The results also show that when a C2V fails, it is down for an average of about 45 hours. These values are computed using outputs from SoSAT not shown here.

Now consider the Black Network functions of the same C2V's. The results show that there is approximately a 5% chance of a network hardware failure and that when one does occur, network capability is down for an average of about 6.2 hours. The 6.2 hour downtime is a function of the repair distributions used on the network hardware components. Given the unlikelihood of network hardware failure and the relatively short duration of such failures, it is unlikely that a meaningful overlap will occur and thus the network hardware will most often be available when needed.

While this conclusion is justified, the purpose of this project is to introduce capabilities for the modeling of network effects into the current logistics simulation. As such, it is prudent to perform additional simulations with altered (perhaps even unrealistic) data in order to test the capability and verify that observed results are in keeping with expectations. Therefore, each of the following batches of results is prepared using altered values of failure rates and/or time to repair distributions for the network and platform hardware.

Running the simulation *without* network effects is functionally equivalent to running it with a perfect network in which there are never any failures of network hardware. The converse is to run the simulation using network effects but to fix it such that the network is never available. Doing so produces the following results.

Table 3-4. Comparison of Perfect with 100% Imperfect Network Hardware.

Platforms	Operational Availability		Difference (%)
	Without Network Effects	With Network Effects	
Top	76.2%	53.3%	-42.96%
C2V	78.0%	34.5%	-126.09%
CAMEL	88.0%	85.7%	-2.68%
FLEX	100.0%	100.0%	0.00%
FRMV	64.0%	22.3%	-187.00%
HEMTT-LHS	76.9%	29.1%	-164.26%
HMMWV-SPT	68.4%	17.3%	-295.38%
HMMWV-C2	67.6%	17.0%	-297.65%
ICV	76.3%	33.7%	-126.41%
M-1	100.0%	100.0%	0.00%
MULE-CM	88.1%	50.7%	-73.77%
MV-E	38.5%	26.9%	-43.12%
MV-T	41.5%	42.6%	2.58%
PLS-Trailer	100.0%	99.9%	-0.10%
Tank-Rack	96.5%	95.5%	-1.05%
UAV1	73.7%	68.1%	-8.22%
UAV2	45.8%	39.0%	-17.44%

This output shows the maximum possible effect the network hardware can have and it is substantial but corresponds to the unrealistic circumstance where all network hardware is inoperable for all time.

In order to characterize the behavior in the intermediate levels of network hardware availability, the following is a plot of the **Top** rollup A_o for varying MTBSA for the network hardware function beginning with the actual FCS data and progressing down to an MTBSA of ½ hour.

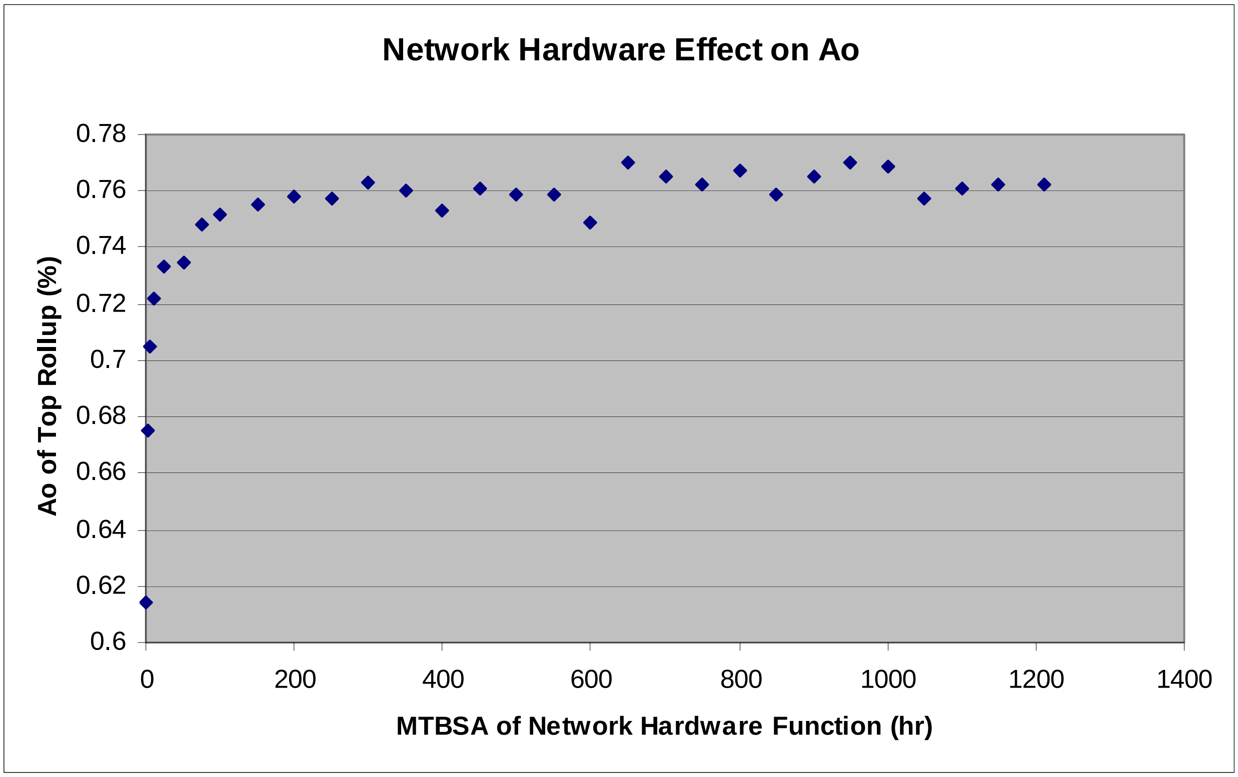

Figure 3-26. Effect of Decreased Network Hardware Reliability on A_o.

This plot shows that A_o is not strongly affected by varying the MTBSA until MTBSA falls below about 150 hours. This is sensible since the total number of operational hours during each replication is 144 (recall there is a 24 hour replenishment period in our 168 hour mission). This reduction in A_o below 150 hours MTBSA is in keeping with expectations that as the network hardware becomes less and less available, the A_o of the BCT will decrease.

4. Conclusions and Future Work

This LDRD project has been a success in the creation of an initial treatment of communications network considerations in the current SoSAT software. An approach has been developed and implemented in a version of SoSAT slated for future release. The results presented in Section 3.3 demonstrated the impacts of the network hardware on operational availability of a notional BCT and were in keeping with expectations.

In the future, additional capabilities will be required to account for non-hardware related network failures. These include considerations such as:

- **Environmental Conditions –** conditions such as storms and unfavorable terrain can degrade network performance.
- **Priority Assignments –** In a combat scenario, not only logistics messages but many different kinds of messages must be transmitted over the network. Because of this, priorities are assigned to the message categories and in particular, logistics messages tend to have low priority.
- **Bandwidth Issues –** As message volume increases during a combat scenario, bandwidth limitations become important. This is when the priority assignments become important.
- **Message Completion Rates –** For any given scenario, there will be some percentage of messages attempted that never reach their recipients.
- **Dependency of Waveforms on Relaying Platforms –** Not all platforms in the BCT are capable of transmitting all waveforms and some waveforms are dependent on intermediate platforms to serve as relays (ex: WIN-T requires a satellite). Therefore, it is necessary that in representing waveform capabilities on platforms, that these dependency requirements be represented.

Current information coming from the FCS program indicates that the force-on-force models in use will be able to provide certain statistical quantities that may in turn be used in SoSAT for these purposes. As an example, statistics are expected that describe the probability that a platform in a particular subnet will be able to communicate with one in another subnet. Additional code will be required in SoSAT in order to make use of these quantities once they are available. The improved reference capabilities described in Section 2.3.4 address the final bullet.

Distribution List

	MS 1001	Roehrig, Steve	06300
	MS 1005	Skocypek, Russell	06340
(10)	MS 1011	Anderson, Dennis J.	06342
(2)	MS 1011	Campbell, James E.	06342
(5)	MS 1011	Cranwell, Robert M.	06343
(2)	MS 1125	Eddy, John P.	06342
	MS 1011	Lawton, Craig R.	06342
	MS 1011	Le, Hai	06342
	MS 1011	Thompson, Bruce M.	06342
	MS 1011	Vander Meer, Robert	06342
	MS 1011	Welch, Kimberly	06342
	MS 1011	Yaklin, Laura	06342
	MS 1005	Nanco, Alan	06345
	MS 0188	D. Chavez, LDRD Office	01011
(2)	MS 9018	Central Technical Files	08944
(2)	MS 0899	Technical Library	09616

Sandia National Laboratories