SANDIA REPORT
SAND2010-8037
Unlimited Release
Printed November 2010

Data-Driven Optimization of Dynamic Reconfigurable Systems of Systems

John P. Eddy and Conrad S. Tucker

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

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

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

SAND2010-8037
Unlimited Release
Printed November 2010

Data-Driven Optimization of Dynamic Reconfigurable Systems of Systems

John P. Eddy and Conrad S. Tucker

System Readiness and Sustainment Technologies Department
P.O. Box 5800, MS1188
Sandia National Laboratories
Albuquerque, NM 87185

Abstract

This report documents the results of a Strategic Partnership (aka University Collaboration) LDRD program between Sandia National Laboratories and the University of Illinois at Urbana-Champagne. The project is titled "Data-Driven Optimization of Dynamic Reconfigurable Systems of Systems" and was conducted during FY 2009 and FY 2010. The purpose of this study was to determine and implement ways to incorporate real-time data mining and information discovery into existing Systems of Systems (SoS) modeling capabilities. Current SoS modeling is typically conducted in an iterative manner in which replications are carried out in order to quantify variation in the simulation results. The expense of many replications for large simulations, especially when considering the need for optimization, sensitivity analysis, and uncertainty quantification, can be prohibitive. In addition, extracting useful information from the resulting large datasets is a challenging task. This work demonstrates methods of identifying trends and other forms of information in datasets that can be used on a wide range of applications such as quantifying the strength of various inputs on outputs, identifying the sources of variation in the simulation, and potentially steering an optimization process for improved efficiency.

Acknowledgements

The Data-Driven Optimization of Dynamic Reconfigurable Systems of Systems team would like to acknowledge the significant support, time, and effort provided to the program by Harrison Kim and Bruce Thompson. Harrison Kim is on faculty at the University of Illinois at Urbana-Champagne and serves as Conrad Tucker's graduate advisor. Bruce Thompson is the Program Manager for this LDRD.

In addition, we would like to acknowledge the considerable support of Kimberly Welch and Craig Lawton of the System Readiness and Sustainment Technologies department at Sandia and that of Dennis Anderson and Alan Nanco of the Military Systems & Analysis Group also at Sandia. Kim and Craig provided technical guidance and subject matter expertise for the Stryker brigade and for the SoSAT application. Dennis and Alan provided financial support through the FCS program as well as technical guidance during the summer of 2009 when Conrad performed some of the work presented in this paper. Finally, we would like to acknowledge the help and support of Russ Skocypec in the Talent Life Cycle organization for his help in arranging and maintaining this collaboration.

Table of Contents

List of Figures

Executive Summary

Sandia's System of Systems (SoS) analysis tools often generate large amounts of data on the order of several gigabytes over many trials. Our current tools include capabilities for visualizing and interpreting the resulting data. However, in all cases, the information and views presented can be considered "low-order", meaning that they are simple plots of the data or other quantities simply calculated from that data. The ability to interpret higher orders of information from such datasets is a high priority need for our analysts to provide the most useful, thorough, and illuminating results to our customers.

The intent of this project is to introduce techniques used in data mining into the suite of tools used to perform SoS analysis and optimization. The large scale data generated by SoS simulation models can be mined to extract hidden, non-trivial, previously unknown patterns within the data set. Such insights will enable analysts to understand the complex Systems interactions of large scale SoS models and help predict the emerging trends and interactions among Systems and Subsystems.

This report documents work completed for the Strategic Partnership LDRD program entitled "Data-Driven Optimization of Dynamic Reconfigurable Systems of Systems." This work shows that a number of data mining techniques can be used to aid in SoS modeling, simulation, and optimization. An example problem shows how such techniques can be used to make predictions about the SoSAT Stryker Brigade model and provides insights into future directions for related work.

Introduction

This section describes the types of problems being addressed by this work and concludes with a discussion of the desired outcomes.

Problem Background

Sandia's System of Systems (SoS) analysis tools often generate large amounts of result data on the order of several gigabytes over many trials. Our current tools include capabilities for visualizing and interpreting the resulting data. However, in all cases, the information and views presented can be considered low-order meaning that they are simple plots of the data or other quantities simply calculated from that data. The ability to interpret higher orders of information from such datasets is a high priority need for our analysts in order to provide the most useful, thorough, and illuminating results to our customers. Examples of such information may include classification of simulation artifacts by properties via clustering, identifying the strength of inputs on outputs, and providing a means of estimating outputs for given input sets.

Project Goals and Objectives

The intent of this project is to introduce the techniques used in data mining into the suite of tools used to perform SoS analysis and optimization. The large scale data generated by SoS simulation models can be mined to extract hidden, non-trivial, previously unknown patterns within the data set. Such insights will enable analysts to understand the complex Systems interactions of large scale SoS models and help predict the emerging trends and interactions among Systems and Subsystems. The fundamental objective is to propose relevant data mining techniques to support and enhance the decision making and strategic planning of SoS model setup and simulation. The long term goal is to establish a proprietary Sandia data mining/machine learning toolset that can meet the needs of a wide array of SoS design problems.

Technical Analysis

System of Systems Modeling

SoS analysis is necessary to understanding the characteristics of large-scale inter-disciplinary problems that involve multiple distributed systems that are embedded in networks at multiple levels and in multiple domains. The tool used to perform SoS analysis in this work is called the System of Systems Application Toolkit (SoSAT). Figure 1 presents an overview of the SoSAT concept.

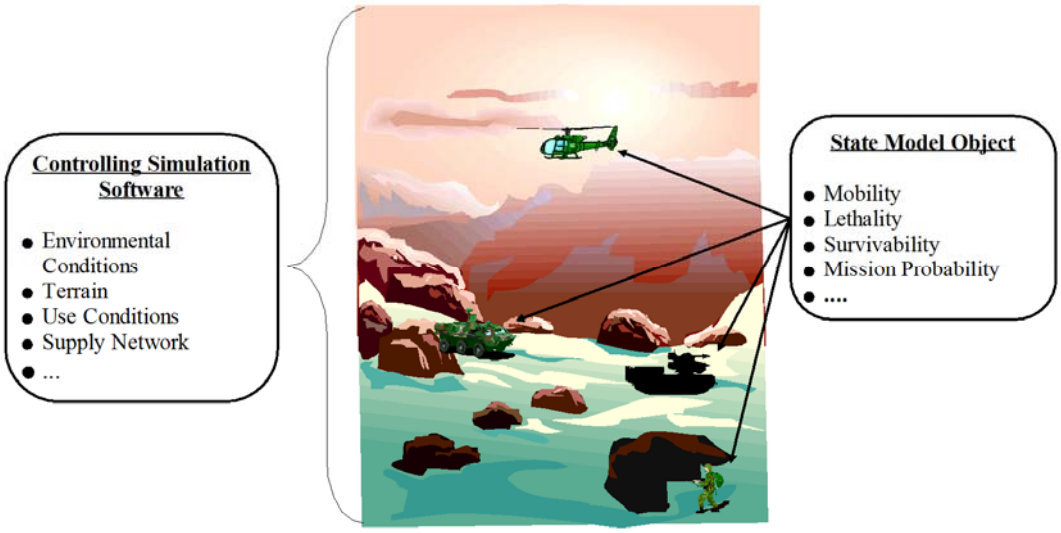

Figure 1: The Multi-System SoSAT Simulation Concept.

SoSAT development was driven by the need to support the Future Combat Systems Brigade Combat Team (FBCT). However, SoSAT has been applied to the design of many varied SoS problems. SoSAT is a time-step stochastic simulation tool designed to model and simulate the multi-echelon operation and support activities projected to be conducted by FBCT. Figure 2 presents a high-level picture of the simulation architecture used in SoSAT. It provides logistics analysts with the ability to define operational and support environments and characterize measures of its performance effectiveness based on multiple trials. SoSAT characterizes sensitivity changes to all platforms, support systems, processes and decision rules as well as vehicle reliability and maintainability (R&M) characteristics. It is designed to be a robust decision-support tool for evaluating the readiness and sustainment of the FBCT to include fuel, water, ammunition and maintenance operations. SoSAT can also take into account external conditions (e.g., storms or extreme terrain) and combat damage. Simulation output results assist the user in identifying platform, as well as SoS level performance and logistics support issues.

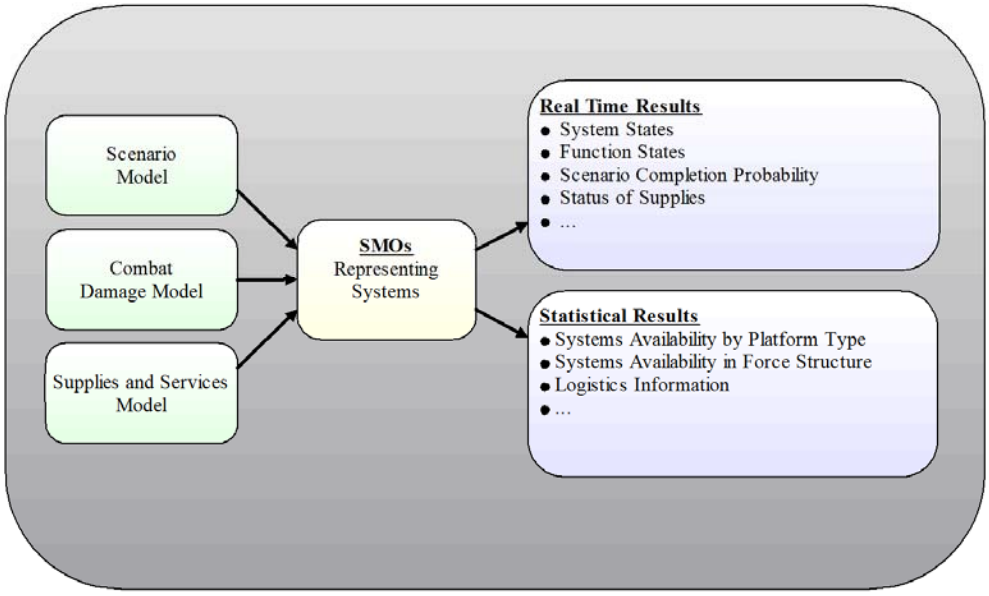

Figure 2: High-Level View of the SoSAT State Model Object Concept.

Key to the multi-system simulation capability has been the development of a State Model Object (SMO) that enables a system, its elements, and its functionality to be encapsulated for use in the simulation. Every system in the simulation is represented by an SMO which has a defined composition of items that help define the system's functionality. SMOs can represent air vehicles, ground vehicles, manufacturing equipment, etc. The systems are the central objects of the model and are the entities that march through the simulation.

The basic structure for modeling a system as an SMO in SoSAT is as follows. A system performs functions (e.g., mobility, communications, sensing, lethality, etc.). Functions are supported by elements of the system, including primary elements (engine, instrumentation, sensors, etc.) and consumables (fuel, ammo, etc.). Elements can fail by normal reliability processes, external conditions (combat damage, external elements—e.g., severe weather, hilly terrain, etc.), and the failure of other systems (e.g., logistics). Failure of an element affects system function. Failure of a function can affect other systems and system availability.

Data Mining

Knowledge Discovery in Databases (KDD) is the non-trivial means of extracting meaningful, hidden patterns within a database [1]. As data extraction and storage capabilities become cheaper and more readily available, tremendous opportunities exist to incorporate the knowledge gained from large databases directly into SoS predictive modeling and design efforts.

In order to fully understand the role of Data Mining in Systems of System modeling, we present the overall methodology that begins with large scale data acquisition, followed by the knowledge discovery process which generates predictive models that can be used in subsequent simulation models. The overall procedure can be represented in Figure 3.

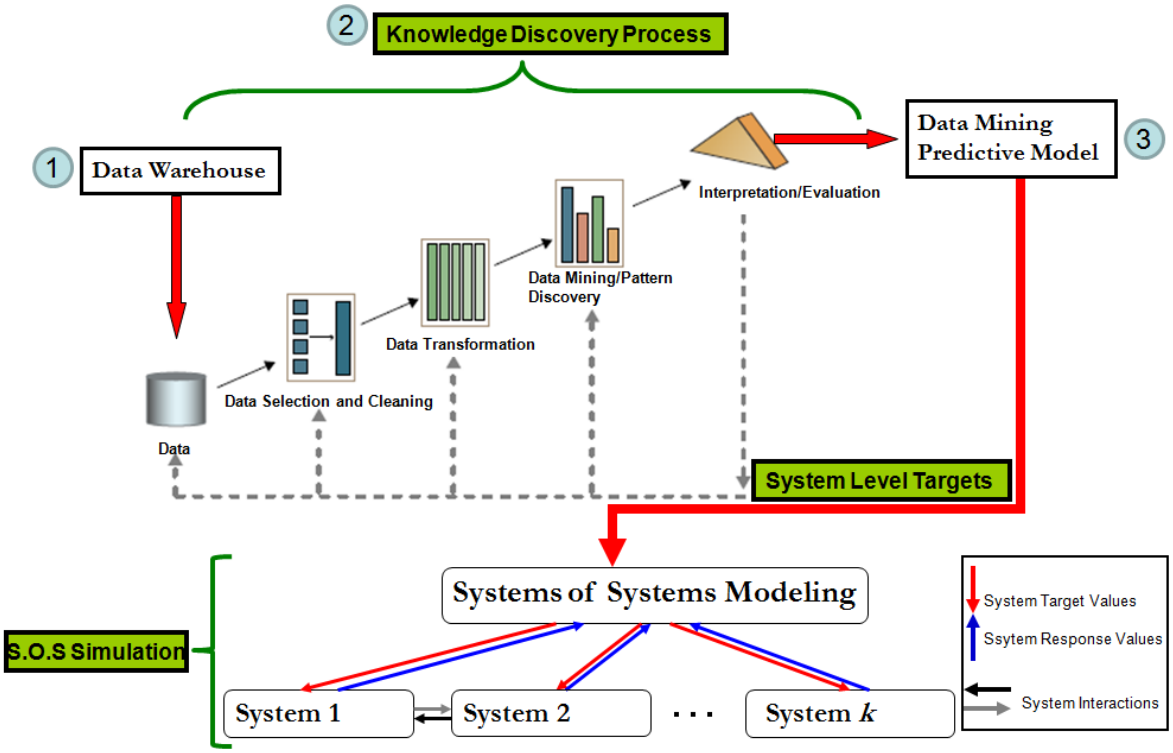

Figure 3: Overall Data Driven Product Design Methodology.

The data driven SoS methodology presented in Figure 3 begins with:

1 *Data Warehouse:* This is where the raw data of previously run simulation models exists in a compact and efficient form. A robust Data Base Management System (DBMS) will enable users to quickly access subsets within the Data Warehouse to be mined.

2 *Knowledge Discovery Process*: This step involves translating the relation acquired from the Database Management System into acceptable forms for the Data Mining Machine learning algorithm. This involves data cleaning (outlier removal, missing value replacement, etc.), data transformation (binning, etc.) and finally employing a Data Mining Algorithm such as Decision Tree Classification [1], Clustering [2], Association Rule Mining [3], to name but a few. The following section presents an overview of some of the data mining techniques employed in analyzing the resulting large scale data generated by SoS models such as those built using SoSAT.

C4.5 Decision Tree Classification:

The C4.5 Decision Tree Classification algorithm is an induction based approach that iteratively partitions the original dataset into subsequent subsets until a homogenous class value (response value) exists in each data subset (or until a minimum threshold is achieved)[1]. The underlying foundation of the algorithm is built upon the concept of *information gain* as a measure of individual system (input variables) predictive power, relative to the class variable (response variable). This can be mathematically represented as:

$$gain(X) = info(D) - info_x(D) \quad (1)$$

Where:

$$info(D) = -\sum_{i=1}^{k} \frac{freq(C_i, D)}{D} log_2\left(\frac{freq(C_i, D)}{D}\right)[bits] \quad (2)$$

$$info_x(D) = \sum_{j=1}^{p} \frac{|D_j|}{|D|} info(D_j) \quad (3)$$

- $\frac{freq(C_i, D)}{D}$ Represents the frequency of a particular class (response) value within the data set D.
- *D* Represents the size of the data set at iteration (q).
- *Dj* Represents a subset of the data set when conditioned on a particular mutually exclusive system value (discrete case) j.
- *X* Represents the current test system for its predictive power, relative to the class (response) variable.

At each iteration, the C4.5 sequentially tests each system (input) variable (*X)* and selects that which maximizes (1) and partitioned the data set D into subsequent data subsets based on the number of mutually exclusive unique values of system variable (*X).*

In SoS modeling, data mining based classification algorithms such as the C4.5 Decision Tree can be used to answer questions such as:

1). *What factors/inputs may be influencing the operational availability of X?* Where X can be any system/subsystem, etc. that has an observable output.

2). *What happens if we increase/decrease these factors/inputs?*

For simulation models that produce discrete output values, Decision Tree Classification techniques such as CART, C4.5, C5.0 can be employed to generate the predictive model [1]. For modeling scenarios involving continuous output values (for example, numeric response such as Operational Availability (A_o)), Regression Tree classification techniques such as the M5 Prime and REP Tree can be employed. These Regression tree techniques have a formulation similar to traditional techniques such as the C4.5, but employ novel evaluation metrics that can handle continuous output values [4, 5].

The M5 Prime Formulation replaces the *Information Gain* metric with the Δerror metric below which enables the model to:

- Handle multivariate linear models, rather than explicit class values
- Handle numeric/nominal attributes, numeric class
- Generate smaller trees

M5 Prime Evaluation Metric:

$$\Delta\text{error} = \text{sd}(\text{D}) - \sum |\text{Di}|/|\text{D}| * \text{sd}(\text{Di}) \quad (4)$$

Where sd() represents the Standard Deviation function of the continuous class values.

In the case of the REP Tree algorithm, the continuous class values are discretized during the iterative decomposition of the data and attributes are evaluated based on the *Information Gain* metric similar to the C4.5 algorithm.

Distance-Based Clustering Algorithms:

There are many well established as data mining clustering algorithms that aim to extract hidden patterns within the raw data set. One well known clustering algorithm is the *k-means* algorithm which has been extended over the years to enhance its efficiency [6]. The underlying mathematical representation can be presented as:

$$f = \sum_{j=1}^{K} \sum_{x \in S_i} \left\| x_i - c_j \right\|^2 \quad (5)$$

Here,

- S_j Represents a cluster of data points. Here, *S* will be defined as all instances in the raw data set and, therefore, S_j would simply be a subset of this.
- c_j Represents the centroid of a cluster S_j.
- x_i Represents a data point existing within a cluster.
- K Represents the total number of clusters (specified a priori by the user).

In SoS modeling, data mining based Clustering algorithms can be used to answer questions such as:

What are the natural patterns/associations that exist between outputs or inputs that can be investigated further?

Association Rule Algorithms:

The Apriority algorithm attempts to find hidden patterns within a given data set by iteratively scanning the database for frequent system-class patterns. *Interesting* patterns that are found must satisfy the anti-monotone Apriori property: *if any length k pattern is not frequent in the database, its length* (*k*+1) *super-pattern can never be frequent* [7]. In the context of SoS modeling, Association Rule Algorithms can be used to determine the frequently occurring input combinations that lead to a particular output response.

Support Vector Machines:

Support Vector Machines (SVMs) is considered a supervised learning algorithm similar to the C4.5 Decision Tree classification algorithm. SVMs use a maximum-separating hyper-plane to partition the instances within the data set to their corresponding class (response) value association [8, 9]. The optimal boundary that maximizes the distance between the class labels and the hyper-plane is found by transforming the original data into a higher order dimensionality space.

Given the training examples $\{x_1, x_2, \ldots x_k, \ldots x_l\}$ and class labels $\{y_1, y_2, \ldots y_k, \ldots y_l\}$, the objective is to minimize over the weights α_k using the quadratic function:

Minimize

$$J = \frac{1}{2}\sum_{hk} y_h y_k \alpha_h \alpha_k (x_h x_k + \lambda \delta_{hk}) - \sum_k \alpha_k \quad (6)$$

Subject to:

$$g1 = 0 \leq \alpha_k \leq C \quad (7)$$

$$g2 = \sum_k \propto_k y_k = 0 \quad (8)$$

(6) sums over all instances of a k-dimensional attribute space. Here, y_k denotes a class label and both λ and C are the soft margin parameters that control the effects of the outliers in training data [9].

Data Trend Mining:

Traditional data acquisition and analysis techniques that have been employed in systems design have relied primarily on static data sets (such as those presented in the previous section). In the context of SoS design problems, the availability of large scale data presents the opportunity to capture emerging systems behavior in a timely and efficient manner. Such capabilities will ultimately enable analysts to quantify the relevance of each system by modeling the time series functionality that may be hidden within the data. The Systems Trend Mining (STM) algorithm aims to address some fundamental challenges of current machine learning techniques being employed in SoS simulation models. The first contribution is a multistage predictive modeling approach that captures changes in systems behavior over time. This is achieved by characterizing emerging system behavior and identifying vital systems,

while classifying non-vital systems as systems obsolete, systems non-critical or systems critical. Due to the interactions that may exist among systems, analysts may be faced with a multi-objective design space that current single objective models do not capture. A time series exponential smoothing technique is then used to forecast future system trend patterns and generate a demand model that reflects emerging systems behavior over time. The overall algorithm flow is represented in Figure 4. The resulting time series decision tree represents the emerging systems relevant to the overall mission objectives.

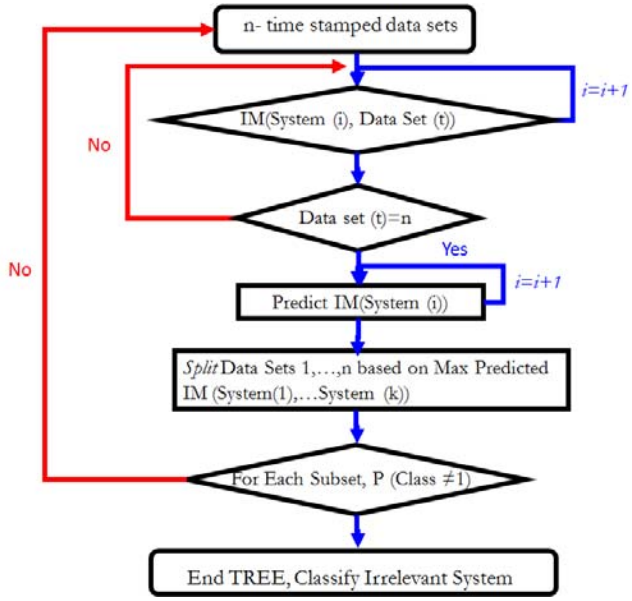

Figure 4: Overall Data Driven Product Design Methodology.

③ *Data Mining Predictive Model*: The resulting Data Mining Predictive Model can serve as an external guide for future large scale SoS simulations. That is, analysts can predict the resulting behavior and interactions of a given large scale SoS model prior to the model actually being executed. This can save a tremendous amount of time and computational resources as System/Subsystem parameters can be adjusted and simulation scenarios updated prior to actual simulation execution. The system level Targets T^C attained from the Data Mining Predictive Model can be represented as:

$$T^C = [A_{1,\ldots,}A_N] \quad (9)$$

Where

- T^C Represents a vector of predicted systems targets based on the particular data mining algorithm employed.
- A_i Represents the specific system interactions that lead to a given class (output) response.

④ *SoS Modeling*: Due to the extensive computational resources required to run a large scale SoS simulation model, system targets $\mathbf{T}^C$ attained from the Data Mining model in step ③ can serve as a

guide to future SoS model simulations. SoS simulations could also use the predicted values from step ③ as constraints in a bi-level optimization model where the resulting SoS model attempts to match the vector of system design targets ($\mathbf{T}^C$) set by the data mining predictive model. When linked with a multi-level optimization model, these targets can be set at the system level objective, while subsystems attempt to share certain design variables/resources [10].

System Level Formulation

$$\textbf{Minimize:} \quad \left\|\mathbf{T}^C - \mathbf{R}^{Eng}\right\|_2^2 + \varepsilon_{\mathbf{R}} + \varepsilon_{\mathbf{y}} \tag{10}$$

$$\textbf{Subject To:} \quad g1: \sum_{k\in K}\left\|\mathbf{R}_k^{Eng} - \mathbf{R}_k^{Eng^L}\right\|_2^2 - \varepsilon_{\mathbf{R}} \le 0$$
$$g2: \sum_{k\in K}\left\|\mathbf{y}_s - \mathbf{y}_{s,k}^{L}\right\|_2^2 - \varepsilon_{\mathbf{y}} \le 0 \tag{11}$$

Here,

- T^C Vector of system targets generated through the data mining predictive model.
- $\mathbf{R}_k^{Eng}$ Engineering response target from the system level, cascaded down to the subsystem level.
- $\mathbf{R}_k^{EngL}$ Performance response target from the subsystem level, cascaded up to the system level.
- $\mathbf{y}_s$ Linking variable at the system level.
- $\mathbf{y}_{s,k}^{L}$ Linking variable value at the engineering sub-system level cascaded up to system level.
- K Subsystem set.
- $\varepsilon_{\mathbf{R}}$ Deviation tolerance between customer performance targets and engineering response.
- $\varepsilon_{\mathbf{y}}$ Deviation tolerance between linking variables.

Subsystem Level Formulation

In the k^{th} subproblem, the design problem is stated as follows.

$$\textbf{Minimize:} \quad f_k + \left\|\mathbf{R}_k^{Eng} - \mathbf{R}_k^{Eng^U}\right\|_2^2 + \left\|\mathbf{y}_{s,k} - \mathbf{y}_s^U\right\|_2^2 \tag{12}$$

$$\textbf{Subject To:} \quad \mathbf{g}_k(\mathbf{x}_k\mathbf{y}_{s,k}) \le \mathbf{0}$$
$$\mathbf{h}_k(\mathbf{x}_k\mathbf{y}_{s,k}) = \mathbf{0} \tag{13}$$

Here,

f_k: Local design objective function (s)

$\mathbf{g}_k$: Inequality design constraints

$\mathbf{h}_k$: Equality design constraints

$\mathbf{R}_k^{EngU}$: Performance response target from the system level, cascaded down to the subsystem level.

$\mathbf{R}_k^{Eng}$: Performance response from the engineering design, i.e., $R^{Eng}=R^{Eng}(x_{Eng})$, (The engineering response R^{Eng} will become R^{EngL} at the system level.)

$\mathbf{y}_s^U$: Linking variable target value cascaded down to the subsystem level

$\mathbf{y}_{s,k}$: Linking variable at the subsystem level

Optimization

The majority of Sandia's SoS analysis tools are for the purpose of analysis of a given input set. The user enters the details of their SoS and a tool such as SoSAT returns information about the operation of the SoS in that given configuration. Of much interest is the ability to go beyond the question of "what will the output be if we change the input as such?" to the question of "what should the input be in order to maximize the output?" This is the question of optimization. To give examples of the way in which optimization could be applied to a SoS, consider the case of a support enterprise for a fleet of deployed systems.

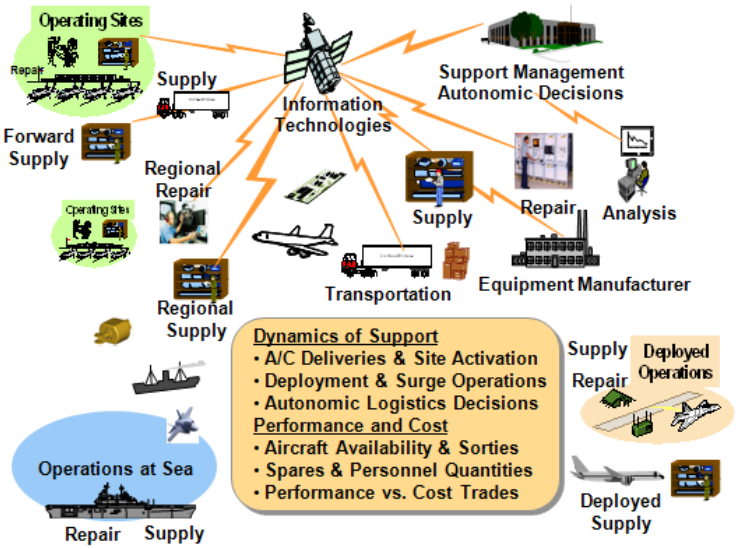

Figure 5: Components of a SoS Support Enterprise.

Analysis of the support enterprise requires consideration of all aspects of the supply chain, repair chain, support equipment, support personnel, etc. There are myriad opportunities for optimization in such a system. Consider the short list of examples below.

-Where should spare parts be stored to minimize downtime due to logistics delays?

-How should spare part inventories be managed in order to minimize downtime due to logistics delays?

-What mix of skills must be kept on hand at points of debarkation in order to minimize downtime due to lack of proper support personnel?

-What new technologies should be introduced in order to manage obsolescence, increase effectiveness, reduce energy requirements, decrease operational costs, etc.?

-What system components should receive reliability improvements in order to minimize downtime due to hardware failures?

And for every question one might ask, there is always the consideration of cost. Beyond asking each of these questions individually, there is a larger picture of the enterprise in which there are relationships between all aspects of the system. For example, reducing cost on inventory may free up funds to add to the staff of support personnel. Increasing the staff of support personnel may make reliability improvements to certain components cost ineffective. In order to learn of and exploit opportunities such as this, a holistic treatment of the enterprise in an optimization problem would be necessary.

Many techniques exist for performing this type of numerical optimization. Each method is well suited to certain classes of problems but none is ideal for all. In the case of an SoS optimization like the example above, a number of features of the problem make optimization challenging. In particular, there are typically many decision variables meaning many degrees of freedom in the model, decision variables are typically discrete, and run-times for the analyses are long. So evaluating a candidate input set is computationally expensive.

There are techniques to mitigate challenges such as these. Examples include relaxations for discrete problems and in order to deal with the case of a computationally expensive simulation analysis, it is common to create surrogates or to create lower fidelity approximations of the simulation. The techniques investigated here have the potential to help with the computational expense. For example, decision trees created using algorithms such as C4.5 can be used as low fidelity approximations relating simulation input values to simulation output values. As such, they may do two things. First, they may serve as surrogate predictors used in optimization. Second, they may show what variables are unimportant to the output thus allowing reduction of the dimensionality of the problem. As another example, classification of the input variables based on the outputs of interest using an algorithm such as the k-means clustering algorithm can serve as a means of reducing the order of the problem by allowing the treatment of multiple variables as a single variable. This will thus also have the effect of reducing the dimensionality of the optimization problem.

Prototype Application

Employing machine learning techniques in the context of systems design and simulation has broad applicability ranging from consumer electronics products such as cell phones [11, 12] to environmentally conscious air purification systems [13]. In the context of SoS, the SoSAT environment is used to

investigate the feasibility of employing machine learning techniques to large scale simulation environments. The dynamic, interconnected nature of the SoSAT simulation model makes it ideal for large scale data mining applications. Here, multiple systems and subsystems are modeled to achieve an overall objective of mission success.

Example Problem

Problem Description

To demonstrate the effectiveness of data mining in the context of SoS design, a large scale SoSAT simulation model was created representing the Stryker Brigade Combat Team. The objective of the case study was to use previously generated SoSAT Stryker Brigade data to generate decision trees that are able to predict output values given input values. Input values are properties of the systems and their components. The properties include the failure rates (FR) and mean times to repair (MTTR) of the components, the repair locations of the components (whether repair can take place in the field, at a repair facility, etc.) and durations and utilizations of the various scenario segments for the SBCT platforms. Scenario segments define what the platform should be doing during a particular timeframe of the simulation. An example would be "Platform A will be in the field from hours 32-48 of the simulation at a utilization rate of 75%" or "Platform A will be in the repair facility from hours 48-72 of the simulation at a utilization rate of 0%", etc. There are a total of 843 inputs used in this example.

Output values are metrics that quantify either:

- the performance of the platforms of the brigade,
- the performance of the echelons of the brigade calculated by "rolling-up" the performance of the platforms within the echelons, or
- the performance of the brigade as a whole calculated by "rolling-up" the performance of all platforms within the brigade.

For the purposes of this example problem, a single output metric is considered. It is the A_o of the various platforms as they execute a 216 hour combat mission with periods of repair and replenishment.

The platforms of the Stryker brigade are shown in Figure 6 below.

Figure 6: SoSAT Striker Brigade used for Data Mining Research Study.

Ideally, for this technique to be useful there would be a large amount of pre-existing data ready for use. That was not the case for this problem. Therefore, 1000 experiments were designed by varying system inputs randomly within their ranges. In the case of FRs and MTTRs, the ranges are a function of the statistical distributions used to define them. In the case of scenario segment durations, the duration of the overall simulation was used to create feasible duration sets. Ranges for Utilizations were chosen to be "reasonable" for the intent of the scenario segment but never vary outside the range of 0 to 100%.

Due to the computational resources required to run such a large scale simulation exercise, a Sandia computing cluster was used to simultaneously execute multiple SoSAT simulations in a timely and efficient manner.

Results

Data mining machine learning techniques enable analysts to answer some fundamental questions regarding large scale simulation models. In the example below, employing Decision Tree classification techniques enable analysts to determine what systems influence the A_o of the M1129 Stryker Mortar Carrier.

The M5 Prime and REP Tree techniques were applied to the SoSAT data collected so that numeric output values of the A_o could be modeled. The A_o can be quantified depending on the branch of the decision tree that is traversed as seen in Figure 7. The order of the System inputs in the tree structure in Figure 7 indicates the magnitude of the system interaction as more critical systems appear higher within the tree. Each partition within the tree in Figure 7 ends with a leaf node which represents the predicted A_o of the M1129 Stryker Mortar Carrier, given the combination of system inputs.

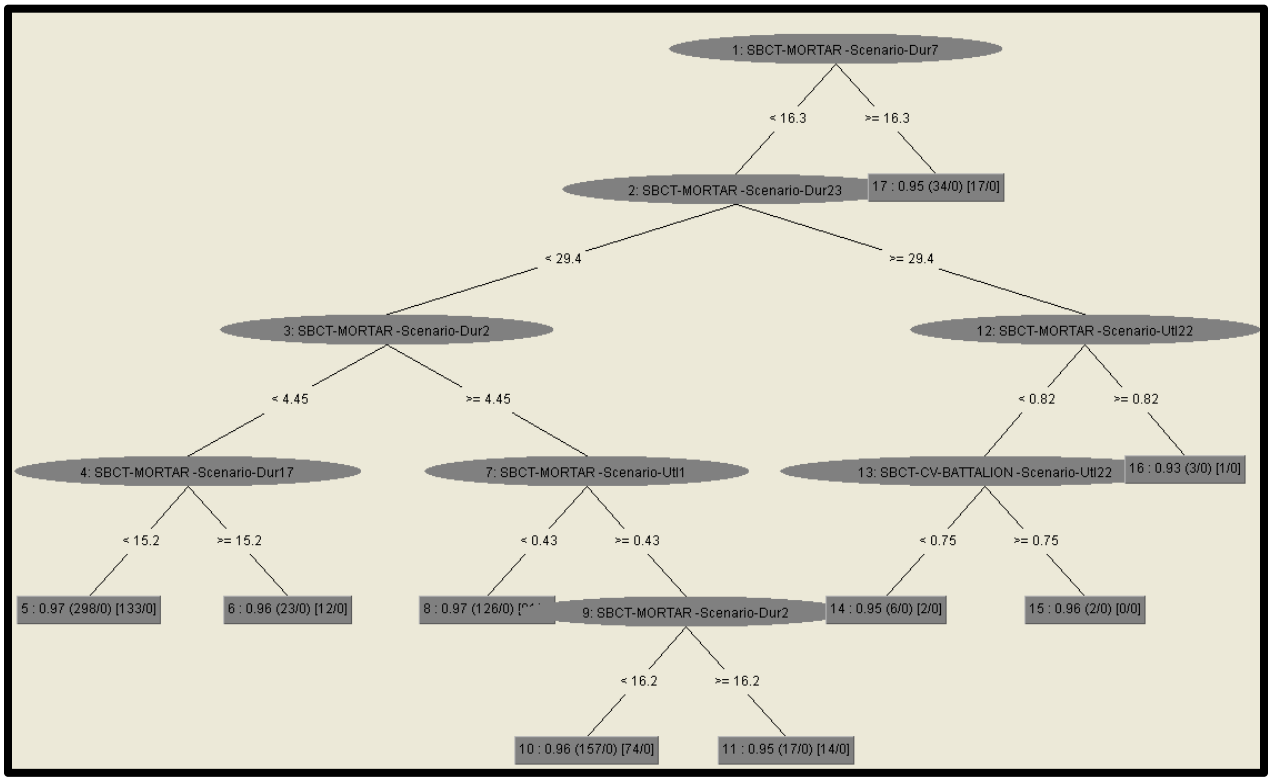

Figure 7: Decision Tree for A_o of the M1129 Stryker.

In addition to predicting the A_o of the M1129 Stryker, the data mining techniques can also reduce the dimensionality of the input space. With more than 800 system inputs at the start of the simulation, the resulting decision tree in Figure 7 reveals that only 8 of those system inputs are relevant to characterizing/predicting the A_o of the M1129 Stryker Mortar Carrier. This insight into the raw simulation data will aid analysts in future simulation setups by allocating strategic decision making resources to those systems that are relevant to the stated objective, rather than the entire system input space which can be of high dimensionality as this example illustrates.

Decision Rules from the resulting Decision Tree in Figure 7 can be interpreted as follows, starting with the right most branch partition.

- IF SBCT-MORTAR-Scenario Dur7>=16.3, THEN A_o of M1129 Stryker Mortar Carrier=0.95 with a Support of 34 (number in parenthesis) and a misclassification of 17 (number in square brackets). The Support of 34 simply refers to the number of data instances (sub set of the original data) that are included in the final node in the tree. This parameter can help decision makers set constraints on the resulting decision tree. For example, they may set a constraint to prune all nodes that have a Support of less than 10 instances so as to minimize the outliers in the model. The misclassification of 17 refers to the number of incorrect output values that were classified,

given a specific path in the tree. This is achieved by testing the actual model with unseen data after the model has been constructed. Once again, decision makers can set the misclassification parameter to be less than a minimum threshold so as to minimize the noise in the model. For example, they may only want to see decision nodes that have a misclassification rate of less than 5%.

- IF SBCT-MORTAR-Scenario Dur7<16.3 AND SBCT-MORTAR-Scenario Dur23>=29.4 AND SBCT-MORTAR-Scenario-Utl22>0.82, THEN A_o of M1129 Stryker Mortar Carrier=0.93 with a Support of 3 and a misclassification of 1.

The remaining decision rules for the entire branch can be acquired in a similar manner as described above. By quantifying the different A_o regions for the M1129 Stryker Mortar, decision makers can focus on areas of combat improvement. For example, if the mission objective was for the A_o of the M1129 Stryker Mortar Carrier to be greater than or equal to 0.95, then decision makers could focus resources on branches such as the second example above where the M1129 Stryker Mortar Carrier has an A_o of 0.93 and make improvements to the system accordingly. In this case, since the relevant inputs are scenario durations (SBCT-MORTAR-Scenario Dur7<16.3, SBCT-MORTAR-Scenario Dur23>=29.4 AND SBCT-MORTAR-Scenario-Utl22>0.82), improvements could be upgrades to the system that make it more survivable and sustainable for operation in those segments.

The results of the Data Mining Decision Tree in Figure 7 help analysts overcome several challenges involving large scale, high dimensional simulation models such as SoSAT. First, as described above, the Decision Tree model allows analysts to narrow down the input space to include only the most *relevant* system inputs that influence/affect the overall mission objective. The second benefit of the Decision Tree model is the ability to quantify the chosen outputs of each of the relevant systems. Analysts can use this information to test hypothesis about the effects of input changes with speed and efficiency. In this way, analysts can use the Decision Tree model as a surrogate analysis model that can help predict the output response avoiding the need to run the simulation for every proposed input change.

A third benefit of this methodology is that it is not computationally expensive. Generating decision trees can typically be done in an amount of time that is orders of magnitude less than the amount of time it takes to run a simulation. Therefore, if the set of interesting inputs or outputs change, new trees can be built quickly from existing data and used. Making predictions given an existing tree is extremely fast and so when used as surrogates trees are a good option for evaluators in an optimization process.

Conclusions and Future Work

In this report we have documented work completed for the Strategic Partnership LDRD program entitled "Data-Driven Optimization of Dynamic Reconfigurable SoS". This work showed that the data mining techniques described have much potential to aid in SoS modeling, simulation, and optimization. The example problem showed how data mining can be used to create a decision tree that can be used to make predictions about the effects of changes to simulation inputs on an enterprise. Future extensions to this work include

- in-depth investigations into the potential to aid in optimization including trials on actual SoS models,
- identification of other aspects of SoS modeling, analysis, and optimization that can benefit from these techniques, and
- development of an application that embodies these capabilities for use by Sandia.

References

1. Quinlan, J.R., *C4.5: Programs for Machine Learning*. 1993: Morgan Kaufmann.
2. Hartigan, J.A. and M.A. Wong *A K-Means Clustering Algorithm. Applied Statistics*, 1979. **28**(1): p. 100-108.
3. Jiao, J. and Y. Zhang, *Product portfolio identification based on association rule mining.* Computer-Aided Design, 2004: p. 149-172.
4. Drobics, M. and J. Himmelbauer, *Creating comprehensible regression models.* Soft Computing - A Fusion of Foundations, Methodologies and Applications, 2007. **11**(5): p. 421-438.
5. Hall, M., et al., *The WEKA data mining software: an update.* SIGKDD Explor. Newsl., 2009. **11**(1): p. 10-18.
6. Tarpey, T., *A parametric k-means algorithm.* Computational Statistics, 2007. **22**: p. 71-89.
7. Agrawal, R. and R. Srikant, *Fast Algorithms for Mining Association Rules in Large Databases*, in *Proceedings of the 20th International Conference on Very Large Data Bases*. 1994, Morgan Kaufmann Publishers Inc.
8. B. Boser, I.Guyon, and V. Vapnik, *A Training Algorithm for Optimal Margin Classifiers.* In Fifth Annual Workshop on Computational Learning Theory, 1992: p. 144-152.
9. Yu, H., J. Yang, and J. Han. ***Classifying Large Data Sets Using SVMs with HierarchicalClusters***. in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2003. Washington, D.C. .
10. Kim, H.M., et al., *Analytical Target Cascading in Automotive Vehicle Design.* Transactions of ASME: Journal of Mechanical Design, 2003. **125**(3): p. 481-489.
11. Tucker, C.S. and H.M. Kim, *Optimal Product Portfolio Formulation by Merging Predictive Data Mining with Multilevel Optimization.* Transactions of ASME: ASME Journal of Mechanical Design, 2008. **130**(4): p. 041103-1-15.
12. Tucker, C.S. and H.M. Kim, *Data-Driven Decision Tree Classification for Product Portfolio Design Optimization.* Journal of Computing and Information Science in Engineering, 2009. **9**(4): p. 041004.
13. Tucker, C., et al., *A RELIEFF Attribute Weighting and X-Means Clustering Methodology for Product Family Optimization.* Engineering Optimization, 2009: p. 1-24.

Distribution List

Internal:

2	MS 1188	John P. Eddy, 06133
1	MS 1188	Dennis J. Anderson, 06114
1	MS 1188	Kimberly M. Welch, 06133
1	MS 1188	Craig Lawton, 06133
1	MS 1188	Bruce M. Thompson, 06133
1	MS0899	Technical Library, 9536 (electronic copy)

External:

1	Harrison M. Kim University of Illinois at Urbana-Champaign 117 Transportation Building 104 South Mathews Avenue Urbana, Illinois 61801
1	Conrad S. Tucker University of Illinois at Urbana-Champaign 117 Transportation Building 104 South Mathews Avenue Urbana, Illinois 61801


Sandia National Laboratories