

T.LOADS ABBREVIATED SYSTEMS ARCHITECTURE

Bob Hamber

1100 23rd Avenue
Expeditionary Systems Division
Naval Facilities Engineering Service Center
Port Hueneme, CA 93041, U.S.A.

ABSTRACT

The Tactical Logistics Distribution System (T.LoADS or TLoaDS) is a powerful and flexible simulation application for assessing current or future tactical or sea-based distribution systems. In its current state of development, it is an analytical model for assessing the pros and cons of new doctrine, distribution techniques, organizational structures, and equipment concepts. It can also be used to find out how to best use available resources to sustain a military force in a wide variety of scenarios. The "System" in TLoaDS refers to its modular architecture and that it is more than just a simulation. It is a suite of analytical study tools. This paper introduces TLoaDS features by an examination of its current software module architecture. It also covers a few of the modules under development to expand TLoaDS scope, depth and range of use, and further improve its ease of use.

1 INTRODUCTION TO TLOADS

1.1 Need

New doctrine, procedures, techniques, organizations, and equipment impose challenges to Navy and Marine Corps expeditionary logistics. These subsystems interact and warrant a model that simulates these dependencies.

The military supply chain is substantially different from commercial supply chains, so commercial supply chain models are inadequate. Three of the key differences are: 1) tactical supply points move, whether ships, combat service support detachments, or combat trains, while warehouses, distribution centers and stores, rarely do; 2) tactical nodes can have chronic shortages of resources, while in the commercial world, more resources are usually acquired as needed; and 3) the tactical distribution system is subject to enemy attack; in the commercial world this is too rare to model.

1.2 Uses

TLoaDS provides a way to qualitatively assess the performance of tactical distribution systems in sustaining the forces ashore from a sea base, port of debarkation, beach support area, or Army or Air Force theater base.

Here are some of the original intended analytical uses of TLoaDS.

- Explore emerging concepts such as Sea-Based Logistics, enhanced combat trains, just in time distribution, and many others
- Assess under what conditions different distribution networks, reorder policies, stock objectives, convoy plans, and many other tactics and techniques are appropriate
- Evaluate alternative concepts for future seabasing ships, cargo handling equipment, transporters, and shipping containers
- Plan the logistic resources required to reliably sustain a given force for a given time in a given threat and environmental scenario
- Determine the odds a given force can be sustained for a given time in a given scenario, with given resources.

With additional input data connectivity and documentation, TLoaDS has many other potential applications including:

- Distributed simulation with combat and strategic logistic models
- Adjudicating war games
- Stimulating command or logistics command and control (c2) systems
- Determining the sustainability of an impending operation or multiple combat courses of action
- Assessing multiple logistics concepts of operation

- Advising on the allocation of distribution resources during an operation to avoid choke points.
- Logistics anchor desk reach-back tool
- Educating logistics students
- Individual or multiplayer educational logistics game.

1.3 Description

At its core, TLoaDS is a supply chain throughput model. The core module is a stochastic “discrete event” simulation, like the commercial supply chain, warehousing and factory industries now turn to for critical problems. It uses the latest commercial-off-the-shelf (COTS) supply chain modeling environment, with custom TLoaDS code components. When it runs, the simulation engine keeps track of how all the individual commodities, orders, shipments, material handlers, and transporters interact with each other and their current environment according to the current process rules. Figure 1 illustrates the general process cycle that occurs between each customer–supplier pair.

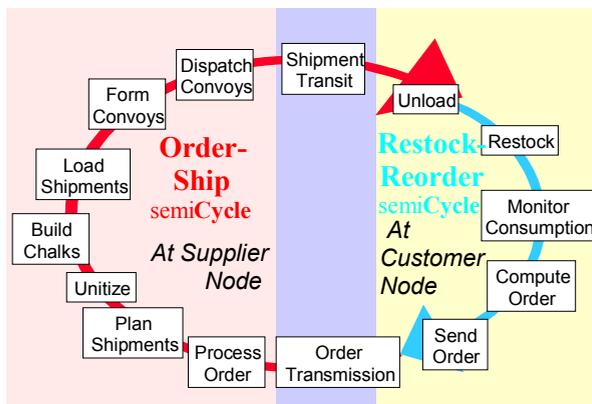


Figure 1: TLoaDS Basic Process Cycle

As TLoaDS runs, it generates a wide variety of outputs that shed far more light into what is going on in the system, than spreadsheet, knowledge base, neural network, or linear programming models do.

Numerous non-core TLoaDS modules described in the body of the paper greatly aid the analyst in managing and processing the vast amount of input and output data involved in a study. These include modules to 1) manage the different simulation levels involved in a study; 2) help prepare and advise the analyst in setting up the model inputs; 3) understand the detailed and overall performance of the distribution system. There is even on-line documentation in the familiar Microsoft Help environment.

For more information on TLoaDS, see our website: <<http://tloads.nfesc.navy.mil>>.

1.4 Sponsorship

TLoaDS development started in 1997 under the sponsorship of Marine Corps Systems Command’s Amphibious Warfare Technology Directorate. The model was originally intended to be just an in-house logistics technology assessment tool. This sponsorship transitioned to the Office of Naval Research (ONR), Code 353 in 1999.

In 1998, ONR, Code 331, sponsored an extension to TLoaDS to simulate processes unique to the intra- and inter-ship operations involved in shipboard cargo handling operations and underway replenishment. This portion of the logistic system gets into the operational level of logistics. Versions of TLoaDS with this functionality, are referred to as C.LoADS or CLoADS (pronounced and understood as “sea-loads”). Unless otherwise specified, this paper applies equally to CLoADS as to TLoaDS including all of the above text.

1.5 Relationship between TLoaDS, CLoADS, X.LoADS and A-LoADS

TLoaDS and CLoADS are application names defined in context of both the current sponsors and initial uses.

TLoaDS is designed to simulate the sustainment and movement of marine air ground task forces (MAGTFs) ashore from ships at sea, as if those ships never run out of supplies. CLoADS is designed to simulate the sustainment of those ships at sea. This includes the shuttle ships resupplying the station ships, and the station ships resupplying the ships that resupply the MAGTF. Figure 2 shows this simplistic application relationship.

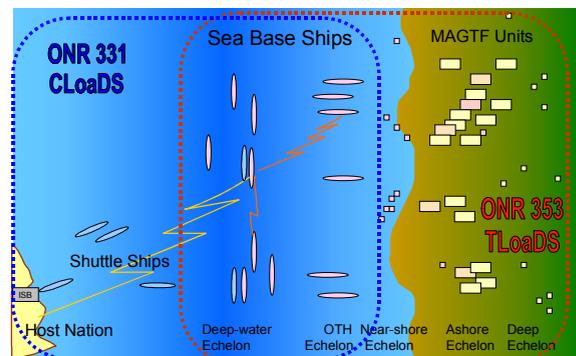


Figure 2: TLoaDS & CLoADS Application Areas

Although in-house, we use A-LoADS to generically refer to the whole general range of TLoaDS, CLoADS, and other envisioned applications, for marketing simplicity and branding benefits, we use TLoaDS as a family brand name. Hence the rest of the paper, including the title, uses TLoaDS in this broad meaning.

2 SOFTWARE ARCHITECTURES

2.1 Formal Architectures

In researching how to specify a overall TLoaDS architecture, we found there are many different types of software architectures.

- The Software Research Institute has many software architecture descriptions. These include simplex, two-tier and three-tier architectures; multiple client-server, mainframe server; and distributive/collaborative enterprise architectures, common object request broker (CORBA) and component object model (COM) architectures. With our current single PC (personal computer) operating mode, most of these architectures don't yet apply to TLoaDS in the overall sense.
- The C4ISR Architecture Framework developed under the Assistant Secretary of Defense C3I and mandated for all C4I systems in DOD, defines three different types of information system software architectures. "The operational architecture view is a description of the tasks and activities, operational elements, and information flows required to accomplish or support a military operation ." "The technical architecture view is the minimal set of rules governing the arrangement, interaction, and interdependencies of the system's parts or elements, whose purpose is to ensure that a conformant system satisfies a specified set of requirements.. The system's architecture view is a description, including graphics, of systems and interconnections providing for, or supporting, warfighting functions. For a domain, the systems architecture view shows how multiple systems link and interoperate, and may describe the internal construction and operations of particular systems within the architecture. Each of these three views are major undertakings. But if one considers TLoaDS the "domain," and its component applications the "systems" then the system architecture view is close to what we have in mind.
- DII-COE (Defense Information Infrastructure - Common Operating Environment) has a four-tier architecture. TLoaDS components could be organized in an analogous way, but this won't convey some of the structural and communication relationships we desire to show
- UML (unified modeling language) defines nine different diagrams, some of which could be considered architectural templates. The deployment diagram appears useful and maybe the component diagram

- ANSI software flowcharting, IDEF, and ERwin are useful process diagram and data relationship methodologies, not architectural templates appropriate for, or outside the scope of this paper.

Most of the above are useful for software design, risk management, and configuration management, but are too involved to be of interest to someone seeking an general understanding of TLoaDS. Since the system architecture appears to be closest to what we have in mind for this paper, but we are sure we are not fulfilling the requirements of the Joint Technical Architecture, we are calling this paper's architecture an abbreviated systems architecture (ASA).

2.2 Abbreviated Systems Architecture Conventions

With the assumption that a formal systems architecture is beyond our project's current resources, we took a clean-sheet approach and created our own architectural diagram (Figure 3) with the following conventions:

- Modules represent either
 - COTS applications, add-ins, plug-ins, or modeling libraries (shaded yellow in Figure 3)
 - Government-off-the-shelf (GOTS) executables, user interactive spreadsheet or database files that run on the COTS applications, add-ins, modeling libraries, or master data files (shaded red). Currently all but one of these is a TLoaDS product (the NIMA product)
 - User created and managed files (shaded green). TLoaDS will provide samples or templates of these files, but the user will modify them with their data.
- From left to right, the modules are generally in order in which they are called upon in the evolution of a simulation project
- From top to bottom, generally the modules below are required for the modules above to run, be used, or created
- Modules that are touching vertically run together. For example the way an ASCII text file needs a text editor (e.g. EDLIN or Notepad) to be created, viewed, modified and saved
- Modules separated by dashed lines transfer data manually via keystroke entry or cut and paste
- Modules separated by dotted lines, write-read data when instructed (manually or automatically) at the end of a study / analysis step
- Modules separated by dot-dash lines communicate via the Extend run time "executive."

Figure 3 corresponds to the “graphics” referred to in the C4ISR Architecture Framework. The next section of the paper corresponds to the “may describe the ... operations of the particular systems within the architecture.”

3 ABBREVIATED SYSTEMS ARCHITECTURE

At the current time, we organize TLoaDS files into 26 modules (Figure 3). Since this is a near term to-be architecture, the use of the present tense in the following descriptions reflects our March 2001 plans for TLoaDS v1.0, to be released in October 2001. All of the 26 modules currently exist in some degree of maturity. The degree of integration may be less that described. For example two modules that are planned to automatically pass data at the beginning or end of some study step, may have to exchange data via a transfer file, or via two transfer files and a simple field name and attribute translation utility.

3.1.1 Operating System Level

The operating system, including device drivers, provides the common CPU, port, motherboard, drive, keyboard, mouse, screen, and printer management functions.

3.1.2 Windows

All TLoaDS modules run in the Microsoft Windows environment, Windows 95 and later.

3.2 Application Level

Application level modules run directly on the operating system. Their core file is an executable (.exe) file. Foundation applications are widely used COTS applications with multiple uses.

3.2.1 Foundation Model Building Application

Extend, by ImagineThat!

Extend is a general purpose modeling application. It can handle continuous, discrete event, or mixed (both) models. Blocks are the key modeling components in Extend. Each block is a reusable component of code. Some blocks perform administrative modeling services (like time management), while others represent parts of a system (e.g., a queue, server, entity creator, process delay, amplifier). Blocks are stored in libraries. Models are built with three main steps.

- Drag block icons from library pallets onto windows call worksheets
- Place the cursor over and input or output connector, and draw a line to one or more connectors on other blocks
- Double click on block icons to open dialog boxes to configure their behavior as appropriate for their locations the model (steps in the process or algorithm)

Many blocks are little models in their own right. In a way, Extend is a distributed simulation environment. Blocks added to the worksheet automatically “register“ to the block manager. The lines drawn between connectors form the basis for their “subscription“ to information. Simulation time management services are provided in the “Executive Block.”. A future article may explore how Extend’s architecture and function parallels the HLA more deeply.

3.2.2 TLoaDS Initial Optimizer Application

TLoaDS Initial-Optimizer (TIO), by Liege System Science Corporation

Liege Inc used their proprietary RapidPro Tool Suite to create a license free optimizer for the front end of the TLoaDS core discrete event module. The TIO integrates

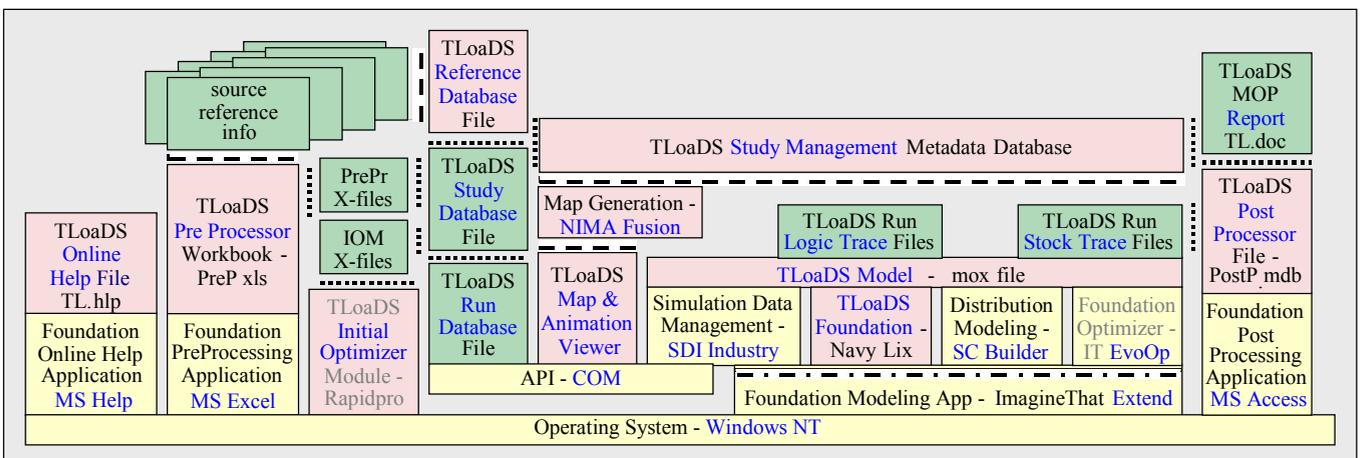


Figure 3: TLoaDS Module Diagram

multiple state-of-the-art optimization techniques including neural network, genetic algorithms, and petri nets.

This is the only custom TLoaDS module to run directly on the operating system.

This custom executable file reads in limited scenario and logistic process data and maximum number of transporters available of each type. The analyst sets some weighting criteria such as the relative importance of minimizing transit time or minimizing transporter attrition. The using petri nets it almost instantaneously computes either the optimal distribution network or optimal kind of transporter. Due to a lack of consideration of queuing and dynamic weather effects its results lack the fidelity of TLoaDS discrete event module (DEM), but it's lightning speed allows the analyst to quickly screen a variety of distribution tactics and select initial values for some critical DEM inputs.

3.2.3 Map Generation Application

NIMAMUSE, by National Imagery and Mapping Agency (NIMA)

This is a set of programs and utilities designed to work with NIMA mapping, charting and geodesy data, and information. We want to use it to prepare raster maps of TLoaDS scenario areas with just the features of interest to the analyst and study customer. The three main steps to do so are:

- Access and prepare NIMA digital data using Raster Importer and Vector Importer programs. Different map and chart features are in different layers
- Merge the chosen layers into a custom made raster map object using the Fusion program. The map can also be saved in Fusion's native "Map Document" format, which preserves the separation of the individual layers for later use with Fusion
- Convert it to any of many popular formats. Foremost will be a format compatible with our map based GUI. We will also want formats for use in reports, briefings, and web pages.

NIMAMUSE has other specialty applications of interest to some users. Perspective Scene builds "from the cockpit window" views analysts may use to decide how to employ logistic resources. Line Of Sight provides terrain analysis for ground-based operations that could be of interest to combat models we hope to federate with.

This module gets data from NIMA source reference information in accordance with the study objectives captured in the Study Management Metadata Database, and pastes it into the Map and Animation GUI. Yet this module runs directly on MS Windows. We cannot follow both our "touching – run together" ASA convention and our "transfer – dashed line" convention when limited to two

dimensions. We have positioned this module in accordance with the latter convention.

3.2.4 Foundation Database Application

Access, by Microsoft

This is the foundation application for the TLoaDS Post Processor.

3.2.5 Foundation Spreadsheet Application

Excel, by Microsoft

This is the foundation application for the TLoaDS Pre Processor and for many source reference files.

3.2.6 Help Application

Windows Help, by Microsoft

This is the help interface common to all Microsoft products, and also used by many others. Topics can be found by Table of Contents, the Index, or a Find/Search function. TLoaDS uses the new hypertext and world wide web capable version. The user can create bookmarks and annotations.

3.3 Intermediate Module Level

Intermediate level modules run directly on one of the foundation applications. They can not run without their corresponding foundation application being launched. In our case, all our intermediate applications run on Extend. Their core software are Extend "blocks."

3.3.1 TLoaDS On-line Help Module

This is a collection of custom .hlp files of TLoaDS documentation. There is one .hlp file for each of the TLoaDS Intermediate Modules.

3.3.2 TLoaDS Pre-Processor Module

This is a collection of Excel spreadsheets we have used for aggregating, disaggregating, and rearranging source reference data and information so it is ready to read into a TLoaDS database.

3.3.3 TLoaDS Map and Animation GUI Module

TLoaDS custom Delphi plug-in, by Denzein Consulting Corporation per NFESC specification and SDI coordination

Our plan for TLoaDS map-based GUI is to use the same GUI used by the Marine Corps in their other PC-based command and control applications -- C2PC by INRI. But all beta users agreed we needed something better than what we had been making due with for years. FY01 project resources and INRI availability were insufficient to

guarantee adequate success, while SDI's subcontractor's proposal looked very promising, affordable, and low risk, we invested in the latter to tide us over. Features include:

- Zoom to mouse-drawn rectangle, drag map around, auto resize to fit window
- Read force positions directly from TLoaDS model database and depict their position overlaid on the map image. Force positions can be changed in the Map GUI model and the changes are saved back to the TLoaDS database
- Drill down into node data by clicking on a node
- Filter map layers and on node attributes
- Animated nodes and transporters move to show each position update
- Transporter animation is shape coded to indicate vehicle type and color coded to indicate loaded condition
- Shipment information can be accessed by pausing the model during animation and clicking on the shipment icon
- Playback animation with a media player like slide control. (Note: This functionality has been delayed.)
- See UTM location of cursor, time and elapsed time in simulation
- The model database includes data fields which allow the user to control the appearance of animated icons representing different types of nodes and kinds of transporters, and individual node labels
- The model database can be viewed and edited without the DEM being open by using a SDI database server plug-in
- Allows the addition of static drawing elements or features to the map image which can be optionally displayed during the animation.

This Delphi windows object will finally provide TLoaDS with professional appearing Run and Laydown screens.

This module communications with the Simulation Data Management Module via a COM API.

3.3.4 Simulation Data Manager

SDI Industry (SDII) by Simulation Dynamics Inc. (SDI)

The first key block in SDII for TLoaDS is the Database Manager block. This block manages all the inputs, key current state parameters, and detailed result data for an individual DEM run. The model database exists as an embedded object within the Database Manager block. Database Manager block provides a link to the Delphi implemented database tool. All data is now held in the Delphi db structure rather than in Extend arrays. The database plug-in includes a GUI for the user to view the data tables in

whatever order and groups he desires. Data can be copied from or pasted in to the tables. Type and kind parameters that can only have defined values are selected by a pick list in each cell to avoid user entry errors. Double clicking a real (versus integer or text) parameter accesses the random distribution setup dialog. Importing and exporting model data can be done interactively or automatically during the initialization and end of a simulation run.

The database add-in stores the data structures in the computers RAM to store the inputs, current state, and results during the initialization of a DEM run. Index (pointer) values are used in place of text variable names to store data efficiently. Since all the data is resident in RAM during the run, the simulation can run much faster than in conventional Extend simulations of the same system.

Other blocks used from this library such as wizards, table displays, table reports, buttons, import and export file controls, and general purpose plotters, are built with the ModL language which is included with Extend .

3.3.5 TLoaDS Runtime Library

Navy Library, by SDI per NFESC

This is a collection of custom Extend block that provide both GUI features and distribution processes not found in the standard SDI Industry blocks or Supply Chain Builder core blocks.

The Run Screen Table Displays buttons open pop-up windows that display current state data. As the DEM runs, when a parameter in an open table display changes in the data manager, it changes in the table display. This is useful for seeing what is happening a key process steps deep inside the simulation. This helps the user understand system behavior and aids in model verification.

- The Current Stock button opens the Dump Status/Commodity Status window. The two radio buttons at the bottom of the window determines which status view is active. When "All Materials at One Unit" is selected, a button at the top will read "Select Unit." Clicking on "Select Unit" will pop up the appropriate pick list
- The Urgency button brings up the Order Processor window. This shows the most urgent materials at each node. (It's blank if all materials are over their reorder points.)
- The "Shipment Q's" button opens the "Loading" window. This displays the creation and loading of individual shipments
- The "Convoys" button opens the "Convoy Display Block" window. This shows the specifications for one convoy rule at one node, and the status as these convoys are being built
- The Run Screen "Pause On" button opens the "Pause Block" window that lets the user make the

simulation pause at each occurrence of four different events (many more are planned). These alerts are a step in the development of TLoaDS on the way to “analyst-in-the-runtime-loop“ “collaborative planning“ (including user approval/denial of simulation feedback)

- The Run Screen “Realism Control” button opens a dialog box where many model fidelity features can be suppressed. (Each bullet covers multiple features that can be individually suppressed)
 - Environmental Conditions: surf, and sea state
 - Resource limits: vehicles, material handling equipment (MHE), fuel for transports, loading spots
 - Attrition: unit strength, transporter, and MHE
 - Equipment Failures: transporters, and MHE
 - Time Delays: issue order delays, material processing delay, transit delays, and transporter refueling
 - Consumption Rate Modifiers: ”factors,“ and “multipliers”
 - Special Supply and Transit Techniques: alternate resupply, limited maximum storage, convoy options, escort options, and night vision.

When starting a study, it is handy to do the first run with all these suppressed so simple system behavior can be observed and nominal idealistic performance boundaries established. As each new run suppresses fewer realistic conditions, the dynamic behavior is more readily understood, while performance generally degrades. Once no fidelity features are suppressed, the analyst can proceed with doing his tuning, optimization, or stochastic runs as he desires.

The customized Wizard blocks that allow the user to step through all the tables related to specifying a new kind of entity are in this library.

The customized Supply Chain Builder blocks are:

- TLoaDS Fill, which decides which kind of container/pallet - and kind of transporter is going to ship each load of material.
- Unitization, which simulates the delay in drawing material from stock and unitizing for shipment and moving it to the “chalk” where it waits for a transporter and MHE to load it.

There are also two custom consumption calculation block completely unlike anything in Supply Chain Builder, or other Extend or SDI libraries.

- Scenario calcs, which look up current scenario conditions from the appropriate scenario tables and set current states such as which types of transporters can access which nodes

- Rate calcs, which look up current scenario conditions and compute the hourly rate of consumption for material at each node.

3.3.6 Supply Chain Modeling Library

Supply Chain Builder by SDI

This new-in-2000 COTS M&S module is a major breakthrough in simulation of supply chains, Six Extend and SDI Industry compatible blocks do most of the work.

- Consumption – simulates consumption of supplier
- Order – generates orders for supplies
- Assign – assigns orders to suppliers
- Fill - plan shipments
- Route - routes shipments to next location
- Supply Chain Manager.

The first five capture supply chain processes. The sixth manages these five functional process blocks (Simulation Dynamics Inc. 2000).

The Route block can simulate delays. We use this many ways.

- Allocate MHE between loading and unloading spots
- The individual times it takes to load or unload different kinds of containers or pallets on different kinds of transporter using different kinds of MHE
- Refuel transporters
- Maintenance delays
- Delay time due to processes and delays between the material being unloaded and when it is available for consumption, reunitionization or reloading.

Last year, TLoaDS required custom versions of the SCB Route block to simulate these delays. Now the Route block has a database lookup feature so these delays can vary by the various types or kinds of resources and inventories.

There is also a special plotter for displaying how inventory stock levels (and other parameters like reorder point, stock objective and safety level) change with time.

Different blocks of the same type are “scoped“ to apply to different sets of inventories. The magic of these blocks is how they automatically figure out which instructions / events / resources / information are to be processed by which block. And how computer processing efficient all this simulation is and therefore how fast these simulations run.

A separate component of SCB is Supply Chain Investigator (SCI).

It is used to investigate the causes of problems with supply chain performance. Once an analyst identifies an inventory of interest (say because it ran out during the run), he sees all relevant information about its current state including stock control parameters, unfilled orders, and unfilled ship-

ments. Then by clicking on a “Looking Upstream“ or “Looking Downstream“ button, the analyst sees relevant information for the supplying inventory, or customer inventories. Often displaying this information will identify the problem. If not, the analyst can proceed further upstream or downstream, or for any given order, he can click on an “order,” , “assign,” or “fill“ cell and see both the relevant process logic and open the dialog box of the corresponding Order, Assign, or Fill Supply Chain block. These dialog boxes hold the corresponding process control settings. The Supply Chain Investigator saves the analyst time hunting for the right table, right record, and field in that table, and the right Supply Chain block to figure out what caused the supply problem.

Supply Chain Investigator communicates with other TLoaDS modules via the COM API.

3.3.7 TLoaDS Post-Processor

TLoaDS second Post Processor Module (PPM), by Pacific Research Group (PRG) per NFESC.

This custom Access-based module reads in four specific DEM output files into its tables and using Visual Basic controls plots the output per the analyst’s desires. The plots are Excel chart object linking and embedded (OLE) objects. The postprocessor facilitates querying, correlating and aggregating large amount of DEM input and output so the analyst can better observe time plots, relative magnitudes, effects and trends.

3.3.8 Core Level

The module at this level is the heart and sole of TLoaDS.

3.3.9 TLoaDS Discrete Event Model

TLoaDS-s-e-c-t.mox, by Analyst

The .mox file is where all the Extend-based discrete event modeling components come together. It stores all the model worksheets, which map where all the library blocks are, the connections between them, and how they operate for a given set of runs supporting a subcase, called a “turn.” For example, when the simulation is run, each process block behaves as set up when the executive block says it has an event to simulate, the executive block records and sorts the next scheduled event output by any process block; and the plotter block reads the parameter stream from blocks its inputs are connected to. Whether running; paused; or after a run has terminated, each button performs its correct specified action. After run termination, each read and write block imports and exports the user specified data set; and each wizard block calls up the right tables in the right order and reads in the right supporting notes, and records the user’s inputs.

The wizards allow the user to step through all the tables related to specifying a new kind of entity. They also allow for the modifying, renaming, and deleting kinds of entities.

3.4 TLoaDS Data Base Level

These custom modules were all developed specifically for TLoaDS.

3.4.1 TLoaDS Study Management Metadata Database

The large number of runs and cases called for in one study of a few issues involving dozen or so dependent parameter sets and the massive amount of output requires a “conceptual framework” to manage the information. This conceptualization is a current topic of study, creativity, and debate within the TLoaDS Project Team. A future paper will explain this construct that will standardize communication between different TLoaDS trainers, users, and customers. At time of writing the working conceptual framework level hierarchy is:

- Program – encompassing all studies that draw on a body of reference data
- Study – for varying the highest level of dependent parameter set
- Experiment – for varying the intermediate level of dependent parameter set
- Case – for varying the lowest level dependent parameter set
- Turn – for varying the tuning and realism parameters
- Run – for varying the random seed number to generate multiple stochastic runs.

The parameter sets currently defined by the conceptual matrix are:

- Equipment kind characteristics
- Mission, enemy, troop, terrain (including weather) (METT) inputs
- CSS tactics and technique parameters
- Allocation of resources to each node.

Just what set of parameters is associated with the program, study, experiment, and case level depends on the use case. The matrix of use case versus parameter sets is under development. With more parameter sets than levels to match them up with, we anticipate adding at least one more level (sub-study, sub-experiment, or subcase).

This metadata database will document which parameter sets are used for each turn, and other metadata such as corresponding preprocessor, run database, discrete event model, trace file, post processor, and case measure of performance report file name, analyst name(s), run control settings, unpaused run time, sim duration, computer speed, amount of random access memory, and operating system.

This database will be useful for tracking case inputs, finding candidate cases for other studies to reuse, and evaluating TLoaDS performance improvements and problems.

3.4.2 The Run Database File

This file is read and written to by the Simulation Data Management Block via a COM API. Input data is read at simulation initialization, or the beginning of each period. The record of current state and end state data is written to the file on run termination.

3.4.3 The Study Database File

This database is generated by the TLoaDS user from the Reference Database (see next module description) to contain the inputs for just the entity types, kinds and individuals that are / maybe needed for a given study. For example, if a study is set in the year 2000 timeframe, the study database has no need to be cluttered up with all the data for all the entities not yet fielded. Or if TLoaDS is being used for a scenario set only on land, the study database can not have all the ship and lighterage characteristic data. And without these entities in the input tables, corresponding records and fields in the output tables will not be created. A screen guides the user through the six steps required.

3.4.4 The Reference Database File

This database contains all the input data prepared for use in TLoaDS. We intend for this to be a centrally maintained master database that all users access to create their study databases from, and load their distinctive input data up to. As users create and characterize new entity types and kinds, this information should be fed back to the Reference Database for other analysts to consider using. Validation and accreditation tables will track which data is approved

4 FUTURE WORK

This architecture should be extended to cover the dozen plus additional modules we are considering to enhance TLoaDS. These include the areas of distributed simulation, distributed learning, database connectivity, two way common operational picture connectivity, intelligent agents, collaborative planning, expert systems, autonomous feedback, statistical analysis of empirical input data, rapid setup, robust model tuning, mission and logistics course of action analysis, results visualization, animation, knowledge management, and study management.

The abbreviated systems architecture needs to be extended to a full systems architecture, and a systems architecture should be done at the component (sub-module) level. Operational and technical architectures need to be done. TLoaDS should be documented in the appropriate unified modeling language diagrams. And producing an architecture that parallels the four tier DII COE architecture would be informative.

5 SUMMARY

Our approach behind developing TLoaDS has always been to integrate the best COTS and GOTS modeling and simulation software, and develop new features to meet our military customers' unique needs. Sophisticated modules such as SDI Industry, and tools such as Supply Chain Builder typify how TLoaDS leverages COTS improvements. Custom modules such as Study Management Module, and the Map and Animation GUI, and screens such as the New Transporter Wizard, and Material Distribution Informer typify how TLoaDS is customized for the needs of analyzing complex military tactical supply chain issues in depth. We have not focused all our efforts on coding the highest fidelity model we could, but have balanced that with the tools the analyst needs to be productive.

ACKNOWLEDGMENTS

Dave Parsons and Henry Bell of Simulations Dynamics Inc reviewed the module functional descriptions.

REFERENCES

- Assistant Secretary of Defense C3I. 1999. DOD Joint Technical Architecture ver 3.0. Available online via <<http://www.afcea.org/education/courses/archfwk2.pdf>>
- Carnegie Mellon University. 2000. STR Technology Descriptions. Available online via <<http://www.sei.cmu.edu/str/descriptions/index.html>>
- Hieb, M. and R. Sprinkle. 2000. Simulation Infrastructure for the DII COE Architecture: The Army Vision. Paper 00F-SIW-035, Fall Simulation Interoperability Workshop. Simulation Interoperability Standards Org.
- Simulations Dynamics Inc. 2000. Supply Chain Training Manual.
- Stytz, M. and S. Banks. 2001. Enhancing the Design and Documentation of High Level Architecture Simulations Using the Unified Modeling Language. Paper 01S-SIW-006, Spring Simulation Interoperability Workshop. Available at <www.sisostds.org>

AUTHOR BIOGRAPHY

BOB HAMBER is responsible for T.LoasDS and C.LoasDS vision, functional requirements, initialization, simulation setup, technical support and training. He is a logistician and systems analyst in the Expeditionary Systems Division at the Naval Facilities Engineering Service Center, Port Hueneme, CA. He has over 20 years experience developing tactical logistics equipment and systems engineering for the Navy and Marine Corps. His E-mail and web addresses are <hamberra@nfesc.navy.mil> and <<http://tloads.nfesc.navy.mil>>.