Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

ARCHITECTURAL CONCEPTS FOR A SYSTEM SIMULATOR FOR CONCURRENT PROTOTYPNG OF EQUIPMENT AND CONTROLS

K. Preston White, Jr.
Ryan Fritz
Stephen Horvath
Carlos Orellana
Jonathan Wohlers

Department of Systems and Information Engineering
University of Virginia
P. O. Box 400747
Charlottesville, VA 22904-4747, U.S.A.

Richard G. Fairbrother
William S. Terry

Distribution Technologies
Lockheed Martin Systems Integration--Owego
1801 State Route 17C
Owego, NY 13827-3998, U.S.A.

ABSTRACT

AutoMod® is a leading discrete-event simulation package widely applied in the modeling and analysis of distribution systems. Included in the AutoMod software suite is the Model Communications Module (MCM), which allows an executing simulation to open socket connections and to send and receive messages via TCP/IP network protocol. In this paper we report on a pilot study which explores the functionality of the MCM. In particular, we develop and implement an architecture that can be used to design, test, verify, and optimize control system software interacting with a discrete-event simulation of the system to be controlled. This architecture supports concurrent engineering of controls and hardware prototypes. Application of this architecture can significantly reduce the duration and cost of development cycles for new equipment and systems. In addition, this architecture can be applied to investigate the feasibility of implementing engineering changes in systems currently deployed.

1 INTRODUCTION

Specializing in the integration of large-scale distribution systems, Lockheed Martin Distribution Technologies (LMDT) is the largest supplier of automation to the United States Postal System (USPS) and the second largest supplier of material-handling equipment worldwide. LMDT also designs and manufacturers electro-mechanical equipment used in its distribution systems integration projects, including a broad portfolio of products for address recognition, material sorting, and material handling for postal and commercial customers. Located in Owego, NY, LMDT is part of the Lockheed Martin Systems Integration organization.

Large postal distribution centers process tens of thousands of letters and packages daily. These facilities receive containers of inbound mail from various locations. Mail is automatically inducted into the system and separated into individual pieces spaced along a conveyor segment. Addresses are optically scanned and identified and pieces are then sorted and binned according to outbound destination. The scale of these systems is impressive, with a typical postal distribution facility comprising several miles of conveyors (White et al., 2001).

Because of the large number and high cost of hardware and software components, and the complexity of integrating these components into an efficient and effect system, modern distribution facilities are capital intensive. LMDT currently employs state-of-the-art, discrete-event simulation technology in an effort to minimize the cost and optimize the overall performance of these systems. Simulation modeling and analysis are used to preview the dynamic behavior of individual components, minor and major subsystems, and the system as a whole, as pieces of mail flow from receiving to shipping. During the design phase of a project, understanding these behaviors is essential in predicting and perfecting eventual system performance. Simulation allows designers to select components and to organize layouts virtually, improving throughput and cycle times by determining the most efficient composition and arrangement of large systems of interacting people and machines. Simulation also is a valuable aid in the management of distribution operations and the reengineering of equipment subsequent to implementation.

2 BACKGROUND

LMDT's primary simulation and modeling tool is AutoMod (Phillips 1998, Rohrer 1999). The AutoMod software suite includes the Model Communications Module (MCM), which provides constructs that allow an executing simulation to interact with various computing entities external to the simulation model. Using Internet protocols and imbedded libraries, a simulation can interact with another application, a model running on a separate workstation, or even the controller for an electro-mechanical system. These capabilities potentially extend the effectiveness of the simulation through the incorporation of customized user interfaces; the implementation of larger, modularized models running over a network of workstations; and development of new test beds to debug and optimize control system software. (AutoSimulations, 1999).

In this paper we report on a study to determine how the capabilities of the MCM can be exploited to develop a platform to design, test, debug, and optimize control system software prior to the development and deployment of hardware prototypes. The study was undertaken by an undergraduate project team at the University of Virginia, supported by professionals at LMDT (Fritz et al., 2002a,b). The ultimate objective of the study is to minimize the cost and shorten the duration of development cycles for prototype systems. In addition, such a platform could be used to investigate the feasibility of implementing engineering changes in systems currently deployed.

Currently, when a new mail distribution product is being developed, the control software cannot be fully tested until the hardware is physically prototyped and available for experimentation. Once the hardware setup is complete, control system tests are then performed manually, using physical objects, such as sample parcels, flats, letters, and mail trays. When malfunctions are encountered, as is inevitable, the controls must be reprogrammed and/or the hardware system physically altered to make appropriate system changes. The effort required to discover and accomplish required alterations is both costly and time consuming and the process of successfully integrating hardware and controls can consume several months and hundreds of man-hours.

If simulation is used to represent the prototype, however, the software that controls the system can be tested and refined before the hardware is fully developed and implemented. In addition, the control engineer can more thoroughly debug and optimize the controller design interacting with the hardware simulation. This is because test cases need not investigated manually, using physical tests, but virtually, using simulation experiments. The simulation can be used to determine quickly whether the system controller handles the virtual entities appropriately for a large array of test scenarios.

3 EXPERIMENTAL SET-UP

To focus this study, we specifically investigated architectural concepts for a system simulator representing an existing LMDT product—the Desktop Tray Tagging System (DTTS). The DTTS is a manually operated version of LMDT's automated mail tray tagging system which lets operators scan, weigh, and automatically assign ground and airline transportation routing tags to mail trays ready for dispatch.

3.1 DTTS Components and Operation

As shown in Figure 1, the DTTS comprises a handheld optical scanner, digital postal scale, and a dedicated computer workstation with monitor, modem, and label printer. When a parcel is placed on the scale and its barcode scanned, parcel weight and identification information (including the destination address) are relayed to the workstation over RS232 interfaces. Control software resident on the computer contacts a remote USPS reservation system via modem, determines and books the best routing for the parcel, and then prints a routing tag to be affixed to the tray. The control software also relays certain information back to electro-mechanical hardware, such as commands to zero the scale. Operation of the DTTS is depicted schematically in Figure 2.

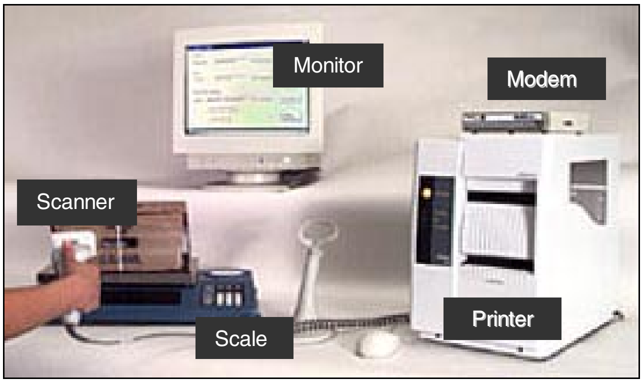

Figure 1: Desktop Tray Tagging System (DTTS)

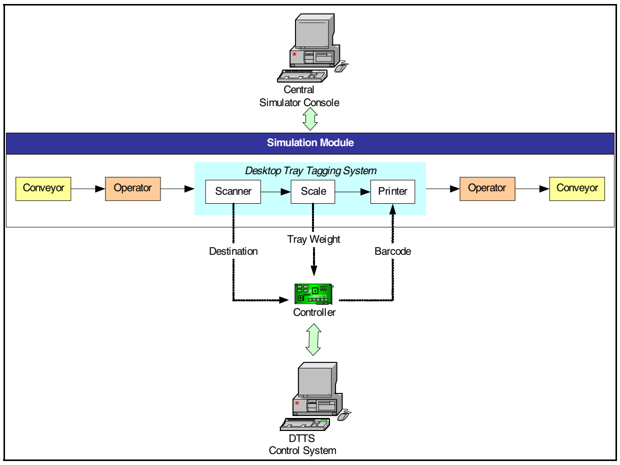

Figure 2: DTTS Simulator Process Flow and Control Schematic

3.2 AutoMod Simulation

A simple AutoMod simulation was written to replace the hardware components of the DTTS. Running on a standard PC, the model simulates the arrival of mail trays at the DTTS queue on a conveyer segment. For each tray processed, the simulation generates output values for the simulated scan and weigh operations. The simulation is paused while these values are relayed to the DTTS controller for parcel route determination and tag printing. The simulation is resumed when the outputs generated by the external control software are returned. From the perspective of the DTTS controller, the simulation output must be indistinguishable from the output of the actual DTTS, allowing the controller to operate as if the hardware had never been replaced. Figure 3 is a screen shot of the simulated DTTS hardware and conveyor segment.

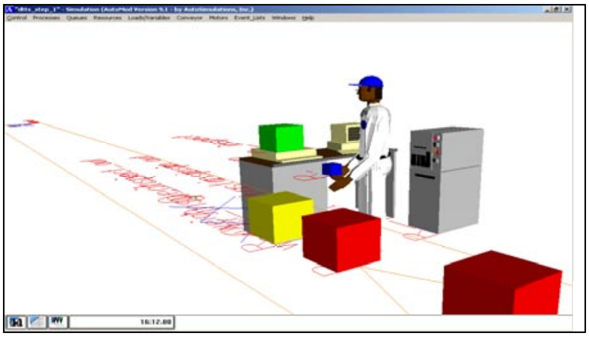

Figure 3: Simulation Model Screenshot

3.3 Communication Components

The most challenging aspect of the simulator design is defining the hardware and software needed to establish connections between the simulation and the DTTS controller. These components relay the appropriate information between the simulation and the three (scanner, scale, and printer) RS-232 data ComPorts of the DTTS controller. Note that on the simulation side, the exchange data are represented as floating-point variables; on controller side, the exchange data are represented in RS-232 (ASCII) protocol (LMDT, 2002). In addition to the physical exchange of signals between the simulation and controller, therefore, the format of the data must be translated such that it can be read and properly interpreted at each end of each of the three communication channels. Alternative methods to achieve this data exchange are discussed in following section.

4 ANALYSIS AND EVALUATION OF COMMUNICATION ALTERNATIVES

After extensive discussions with the client and several hardware and software vendors, the team generated four alternatives for communications between the simulation and the DTTS controller: (1) using software to convert TCP/IP protocol to RS-232 protocol, (2) using hardware (a terminal concentrator) to convert TCP/IP protocol to RS-232 protocol, (3) spooling data to the ComPorts via text files, and (4) spooling data directly to ComPorts. The first two alternatives exploit the functionality of the AutoMod MCM to open socket connections and to send and receive messages via TCP/IP protocol, while the last two alternatives do not.

4.1 Alternative 1: TCP/IP to RS-232 in Software

The first alternative employed uses MCM functions embedded in the simulation to exchange data with a PC-resident software package that converts TCP/IP data to RS-232 data. The software, installed on the simulation computer, accepts information through designated IP addresses assigned on the local network. During the software configuration, each IP address is assigned to a designated serial port on the simulation computer. The three ports and IP addresses of this computer correspond to the scale, printer, and scanner ports of the DTTS controller. Thus, a gateway is created for each type of information to pass from the AutoMod simulation to the correct port of the DTTS controller.

4.2 Alternative 2: TCP/IP to RS-232 in Hardware

The second alternative is similar to the first, except that instead of using internal software to convert the data from TCP/IP to RS-232, an external device (a terminal concentrator) is used. Once the terminal concentrator is connected to the Internet and an IP address is assigned to it, the device is configured such that three additional IP addresses on the local network correspond to the three RS-232 inputs of the terminal concentrator. The three RS-232 ports on the terminal concentrator are then connected to the corresponding RS-232 ports on the DTTS controller.

4.3 Alternative 3: Spool Data to Port via Text Files

This alternative does not require the use of MCM or a local network. Instead, this alternative uses standard read and write functions to transfer data from the simulation to intermediate text files stored on the simulation computer. Once the information is written to the files, the data can be spooled to and from the ports on the simulation computer and then sent through the RS-232 cables to the corresponding ports of the DTTS Computer. This alternative uses three different files, instead of just one central file, so that data can be more easily managed when it is being sent to and from each port. With three different files, each file can be uniquely associated with the three different hardware components of the DTTS controller. For example, the simulation can send each mail tray’s weight to “scale.txt”, and ensure that it is spooled to the correct port with no additional information necessary. Each file can also be cleared every time the information is spooled through the port, so only the current data exists in the file. Clearing the file will create less complexity when spooling the data to and from the ports.

4.4 Alternative 4: Spool Data Directly to ComPort

The fourth alternative is similar to third, in that it does not rely on MCM or a local area network. This last alternative is based on AutoMod's capability to call upon custom C code in the control logic code of the simulation. Therefore, a C function could be called to spool the data directly to the appropriate port. This alternative does not use text files to handle the three different sources of data (simulated scanner, scale, and printer) because the C function can easily differentiate the data by adding additional parameters to the function call. For example, a function could be written in C named *SpoolToComPort,* and called by AutoMod in the control logic every time a simulated mail tray's weight and barcode is received. The function could take two parameters, *DataString* and *ComPortNum*. *DataString* would be a string and could represent either the weight, barcode or printer data that is used in the model. *ComPortNum* would be an integer and represent the port number that is assigned to the scale, scanner, and printer on the AutoMod computer. Using this custom C function would ensure that the appropriate data reaches the desired port on the DTTS computer.

4.5 Evaluation of Alternatives

LMDT established three principal criteria for evaluating the desired simulator: flexibility, simplicity, and cost. Since each new product development environment is potentially unique, LMDT would best be served by a flexible solution that supports both Internet and RS-232 communications among distributed simulations, controllers, and other devices. For the same reason, LMDT desires a simple plug-and-play solution that requires a minimum of custom coding and a straightforward installation for different products. Finally, cost should be low, including both the initial fixed investment in any commercial hardware and software incorporated in the simulation architecture, as well as the recurring investment of professional time required to implement the architecture for any specific product line.

Based on these criteria, the team opted to eliminate the third and fourth alternatives from further consideration. While these alternatives have minimal fixed costs (requiring only the purchase and installation of additional serial ports and standard cabling), neither provides the desired flexibility of Internet connectivity. In addition, these alternatives rely heavily on custom coding and do not appear to meet the desire for a plug-and-play solution. While both the first and second alternatives exploit the functionality of the MCM for Internet connectivity and distributed simulation, if viable the second alternative appears to provide the greater simplicity. Even though this alternative has the highest fixed costs (requiring the purchase of a terminal concentrator and specialized cabling), these investments are relatively modest. Moreover, a hardware solution appeared to be by far the best prospect for having few (if any) continuing costs, because of its virtually plug-and-play implementation. As a side benefit, the second alternative would also provide insight and experience using the MCM that could be exploited in other applications.

5 IMPLEMENTATION AND RESULTS

Implementation of the simulator architecture for the DTTS was achieved in three consecutive development and validation phases. First, the conveyor segment and all of the components of the DTTS (including the controller) were modeled in a single simulation and this simulation was verified. Second, the initial simulation model was decomposed into two sub-models, one representing the controller and the other representing the conveyor segment and all of the remaining DTTS components. The controller simulation model was removed to a second workstation and network connections were established between the two platforms running the two sub-models. The purpose of this step was to ensure that MCM would function properly, with the sub-models communicating over the network via TCP/IP and the distributed simulation yielding the same results as the initial, unified simulation. This phase also was completed successfully and the distributed simulation was verified.

In the final and most challenging phase of the implementation, the controller sub-model was replaced by the DTTS controller using the previously determined hardware communication alternative. A trade study of terminal concentrators found that many companies did not carry products that satisfied the data conversion requirements and, indeed, were unaware of such a connection setup. In the end, per the recommendation of LMDT and confirmation of the supplier tech support representatives, the study concluded that the Digi International PortServer TS-4 was an appropriate conversion device. Upon receiving the TS-4, the team undertook a four-step implementation. The ultimate connection setup is illustrated in the schematic in Figure 4.

Step one consisted of determining how to assign IP addresses the RJ-45 (Ethernet) and RS-232 ports on the TS-4. UVA's Network Administrator assigned four static IP addresses to work with for the remainder of the project. Following the configuration instructions, the TS-4 was setup and available for communication over the Internet. The team then mapped the TS-4's serial ports to the ports on the computer housing the DTTS controller.

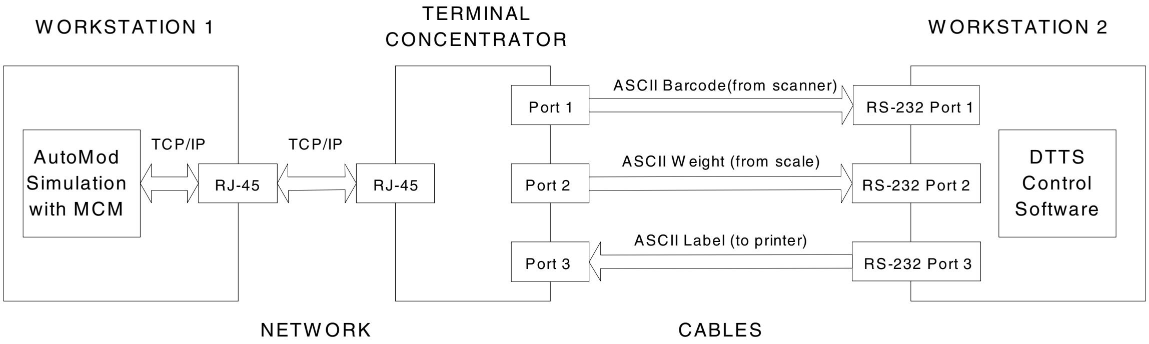

Figure 4: Schematic of the Hardware Communication Alternative

Step two involved testing the IP addresses of the TS-4 ports and their connections to the DTTS controller's serial ports. Using Telnet on the simulation computer, ASCII strings were created from a keyboard and sent over the network, through the TS-4 ports, and finally to the ports on the DTTS controller. HyperTerminal was used to monitor all incoming traffic on the designated ports. The result was a successful connection and data transfer, culminating with HyperTerminal on the DTTS controller computer displaying the test ASCII strings created on the simulation computer.

Having confirmed that the TS-4 could successfully convert TCP/IP data to serial data, step three was to test the DTTS controller software's ability to read simulated ASCII strings. Again using Telnet on simulation computer, ASCII strings representing DTTS' peripheral's signals (scanner and scale) were sent over the network to the controller. The controller successfully recognized the strings as though these had come the actual peripherals themselves and instructed the DTTS label printer to print destination and routing tags from simulated signals.

In the fourth and final step, the team replaced Telnet as the source of ASCII strings with the simulation output and attempted to send the same peripheral signals over the connection to the DTTS controller. This final step exposed a problem with the MCM to TS-4 interface that prevented the transfer from taking place. MCM requires the specification of socket connections in addition to the designated IP addresses, which caused a conflict. The simulation had difficulty opening sockets with the TS-4 and could not make a proper connection. We determined that the underlying cause of this problem is the UTF8 protocol used when exporting data through MCM and the apparent incompatibility of the UTF8 with the protocols normally anticipated by the TS-4 for data transfer. This prohibited communication between the simulation and the DTTS controller. At the time of this writing, we are working actively with Digi technical specialists to resolve this final problem.

6 CONCLUSIONS AND RECOMMENDATIONS

AutoMod is recognized as an appropriate tool for the modeling and analysis of mail distribution systems. It is a comprehensive simulation package with excellent graphical output. In this study, we successfully confirmed that the AutoMod MCM performs as advertised, providing a straightforward means of implementing parallel, distributed simulations using TCP/IP communications, both on a single workstation and across multiple platforms.

At the time of this writing, we are unable to draw any firm conclusions as to the best channel for linking an AutoMod simulation to an external device which communicates exclusively via RS-232 inputs and outputs. We have argued the merits of an architecture which employs the proven Internet connectivity provided by the MCM, together with a hardware device that translates between TCP/IP and ASCII. This is an attractive alternative, principally because of its potential for a nearly plug-and-play setup. In addition, this alternative permits the distributed AutoMod simulation of a system and external devices via TCP/IP connections, as a useful intermediate step in system integration.

However, we have yet to successfully implement this approach. We have isolated the remaining obstacle to UTF8 communications between the simulation and terminal concentrator, however, and continue to work to fit the final piece of the puzzle. We remain positive about resolving this issue over the next several months and hope to report a successful demonstration by the time of the Conference. We also have begun to explore several alternate routes which may provide feasible solutions to the problems encountered in this study. We are particularly interested in applying Windows-based communication tools in conjunction with a VBA and ActiveX compatible simulation package, such as Arena.

REFERENCES

AutoSimulations. 1999. *Model Communications User's Guide*. Bountiful, Utah: AutoSimulations, Inc.

Fritz, R., Horvath, S., Orellana, C., Wohlers, J., White, K. P., Fairbrother, R., and Terry, W. 2002a. Real-Time Control of a Discrete-Event Simulation using an External Controller: A Feasibility Study with Application to Mail Distribution Systems. In *Proceedings of the 2002 IEEE Systems Engineering and Information Engineering Design Symposium*, ed. K. P. White, Jr., Charlottesville, VA: University of Virginia.

Fritz, R., Horvath, S., Orellana, C., Wohlers, J., White, K. P., Fairbrother, R., and Terry, W. 2002b. Control of a Discrete-Event Simulation using an External Controller. In *Proceedings of the 2002 USMA Systems Engineering Capstone Conference*, West Point, NY.

LMDT. 2002. *DTTS Serial Interface Specification: A supplement to the DTTS System, Installation and Users Guide.* Owego, NY: Lockheed-Martin Distribution Technologies.

Phillips, T. 1998. AutoMod by AutoSimulations. In *Proceedings of the 1998 Winter Simulation Conference*, ed. D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, 213-218. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Rohrer, M. 1999. AutoMod Product Suite Tutorial. In *Proceedings of the 1999 Winter Simulation Conference*, ed. P.A. Farrington, H. B. Nembhard, D. TG. Sturrock, and G. W. Evans, 220-226. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

White, K. P., Jr., Barney, B., Keller, S., Schwieters, R., Villasenor, J., Terry, W., Fairbrother, R. G., and Saxton, R.D. (2001) An object-oriented paradigm for simulating postal distribution centers. In *Proceedings of the 2001 Winter Simulation Conference*, ed. Peters, B.A., Smith, J.A., Medeiros, D.J., and Rohrer, M.W., 1007-1012. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

ACKNOWLEDGMENTS

This paper is a substantially revised and updated version of Fritz et al. (2002a).

AUTHOR BIOGRAPHIES

K. PRESTON WHITE, JR., is Professor of Systems and Information Engineering (SIE) at the University of Virginia (UVA). He received the B.S.E., M.S., and Ph.D. degrees from Duke University. He has held faculty appointments at Polytechnic University and Carnegie-Mellon University and served as Distinguished Visiting Professor at Newport News Shipbuilding and at SEMATECH. He is a member of INFORMS and INCOSE and a senior member of IEEE and IIE. He sits on the Board of VMASC and represents IEEE/SMC on the WSC Board.

RYAN FRITZ is a consultant for Accenture in Northern Virginia. Originally from Suffern, NY, he received the B.S. in SIE from UVA in 2002, with a concentration in Management Systems.

STEPHEN HORVATH was born and raised in Fairfax, VA. He was graduated from UVA in 2002, with a BS in SIE and a concentration in Computer and Information Systems. He is currently employed as an IT strategy consultant for Booz Allen Hamilton in Northern Virginia.

CARLOS ORELLANA is originally from San Salvador, El Salvador. He received the B.S. in SIE from UVA in 2002, with a minor in economics. He is currently working as an investment banker for Rothschild NA in New York City.

JONATHAN WOHLERS is an IT consultant with Accenture and lives in Arlington, VA. Originally from Weston, MA, he received the B.S. in Systems and Information Engineering from the University of Virginia in 2002, with a minor in Computer Science.

RICHARD G. FAIRBROTHER is Engineer/Scientist with Lockheed Martin Systems Integration, Owego, NY. Mr. Fairbrother has been designing, developing and analyzing simulations for Lockheed Martin Distribution Technologies the last five years. During this period, he has worked on projects for the United States Postal Service, the Royal Mail (UK), AusPost (Australia), as well as a number of commercial material handling efforts. Mr. Fairbrother's current focus areas are the creation of a Simulation Development Process for Lockheed Martin Distribution Technologies and the exploitation of AutoMod's ActiveX interface with an emphasis on integrating simulations into a multi-application environment.

WILLIAM S. TERRY is the Technology Director for Lockheed Martin Distribution Technologies in Owego, NY. He received a B.S. in Electrical Engineering/Computer Science from Clarkson University and a M.S. in Computer Engineering from Syracuse University. He has worked as a system engineer or system architect on a variety of complex system development programs mostly related to supply chain automation and defense systems.