SYSTEMS INTEGRATION REVIEW: REQUIREMENTS TRACEABILITY, FUNCTIONAL ANALYSIS, AND DESIGN EVALUATION ON THE ILLINOIS DEPARTMENT OF TRANSPORTATION POSITIVE TRAIN CONTROL SYSTEM

Student team: Ryan P. Griswold, Thomas Lai, Chris W. Pritcher, Dong Shao

Faculty Advisors: Garrick E. Louis
Department of Systems and Information Engineering

Client Advisors: Michael Hertz and Stephen Osborne
Lockheed Martin
Naval Electronics & Surveillance Systems-Undersea Systems
Manassas, VA
E-mail michael.hertz@lmco.com or stephen.osborne@lmco.com

KEYWORDS: Positive Train Control, Systems Life-Cycle, Requirements Tracing, Functional Analysis/Modeling, Design Evaluation

ABSTRACT

In a world where demand mandates the development of faster and more reliable transportation systems, Positive Train Control is expedient. Lockheed Martin Federal Systems in Manassas, VA is developing such a system for the Illinois Department of Transportation. This project involves the support of their development effort through three major steps: Requirements Tracing, Functional Analysis, and Design Evaluation.

INTRODUCTION

As today’s engineering projects continue to expand in size, cost, and complexity, the need for a standardized methodology is essential to the development of effective solutions that meet the client’s needs. A number of different systems engineering design processes have been developed and adopted by both the public and private sectors in the hopes of gaining a strong business advantage. Inherent to each of these design processes is requirements management, functional analysis, and design evaluation. The purpose of this project was to perform these steps to assist Lockheed Martin in its implementation of a Positive Train Control System for the Illinois Department of Transportation.

Positive Train Control

In the United States, railroads currently carry over 20 million passengers and nearly 2 billion tons of freight per year and provide a cost-effective means of transportation. The U.S. railroad infrastructure, however, could be the source of significant productivity gains if advanced systems and technologies were implemented to improve safety and system effectiveness (Pellegrino, 1). Current methods of train control have several major flaws:

- No automatic backup intervention to human operator functions
- No positive enforcement of train movement authorities
- No centralized availability of real time train position and status data for safety monitoring and intervention
- Inefficient scheduling and control of freight/passenger traffic
- Fixed block signaling system that allows trains to proceed only to predetermined waypoints
- Signal and crossing gate control that is inadequate for high speed traffic

The goal of the Positive Train Control (PTC) System is to be cost-effective, improve safety, and address the deficiencies found in current train control systems. PTC is based on the same fundamental concept as other train control systems, that railway safety requires absolute separation between trains. The difference is that PTC uses positive train separation and predictive enforcement to achieve a significant improvement in functionality over current train control systems. Positive train separation utilizes dynamic rather than fixed block boundaries. In a fixed block type of system, a train cannot enter into an area of track until the train in front has cleared a distant checkpoint. This reduces the number of trains on the track at a given time and reduces the maximum speed trains may travel. Through the use of precise train locations and real time calculations of safe braking distances, a dynamic train separation system is made possible. This type of system enables safe train separation across a moving boundary, allowing two trains to move in the
same block while maintaining the minimum safe distance between the trains. The PTC system also boasts fundamental changes in tracking and monitoring, allowing for predictive enforcement. Tracking trains, in addition to railway crews and track equipment, allows the system to enforce regulations and restrictions before any violations occur.

The estimated cost per train related fatality is $2.7 million and $250,000 per hazardous material spill (Pellegrino, 4). By implementing a new train control system which addresses current train control system deficiencies, it would not only help reduce injuries and deaths related to passengers and bystanders, but also reduce much of the cost and delays associated with train accidents and repairs. Simply by reducing the number of accidents a train infrastructure experiences, extraordinary gains in productivity and travel efficiency can be realized.

Office Server

The IDOT PTC Office Server is one of the five major segments of the IDOT PTC Architecture and was the focus of our work. The server’s purpose is to manage overall control of the PTC System. It interfaces with trains, wayside units, work vehicles, and the dispatch unit to derive locations, authorities, and restrictions. The Server will maintain route characteristics and speed limit files for transmission to the lead locomotive of trains in addition to ensuring that switches and signals comply with authorities and restrictions.

CORE

Currently, there are hundreds of toolsets on the market to assist systems engineers in designing new products. To perform our work, our Capstone team decided that using one of these tools would significantly increase our productivity. CORE 3.1 was selected for its ability to combine requirements management and functional analysis capabilities. CORE is comprised of classes (requirements, functions, issues) and relationships between instances of these classes. A class is used to represent the major elements of the design process, such as components, functions, interfaces, and data elements. Each class has many attributes that allows the user to fully describe it. The power of CORE really becomes apparent when discussing how relations are created between instances of a class. In the design process there are countless relationships between classes and through CORE, these relationships are tracked, defined, and easily viewed. Another benefit that CORE provides to the design team is the ability to display the system in different views such as hierarchies, ER diagrams, FFBDs, or an N2 diagrams.

SYSTEMS ENGINEERING LIFE-CYCLE

A systems engineering life-cycle provides a framework for the organization and structure of the project design effort. A disciplined top-down life-cycle oriented and integrated process must be followed if cost-effective systems are to be developed and maintained (Blanchard and Fabrycky, 1). This process takes a client’s needs and objects and translates them into an effective solution. There is no one set process that has been adopted by all systems engineers; instead, most corporations and schools have developed their own models. These models are usually very similar and often only differ in how they segment the process. A well known life-cycle model was developed by Andrew Sage, and can be viewed in Figure 1. To satisfy the first two blocks of this model, our team broke them down into Requirements Management, Functional Analysis, and Design Evaluation.

![Figure 1. Systems Engineering Life Cycle Model (Sage, 57).](image)

Requirements Management

One major component of the Systems Engineering Life-Cycle is Requirements Management (RM). This process consists of four main steps which are repeated iteratively. They are requirement elicitation, requirement analysis, requirement tracking, and requirement verification. Requirement elicitation deals with eliciting customer needs. Requirement analysis is the process of interpreting customer needs and deriving explicit requirements that are more specific to a particular part of a project. Requirement tracking involves continuous negotiation within a project team regarding conflicting or changing objectives. Requirement verification embodies the procedures for...
determining whether or not a project design complies with a designated set of requirements.

**Functional Analysis**

Before a design team can develop a physical architecture to meet client needs, the functionality of the system must clearly be defined. Generally speaking, this includes what the system must do, how well it will do it, and what constraints will impact the design (DOD, 45). Functions are discrete actions that the system must accomplish. For example, a function of an automobile is to brake. Functional analysis begins by identifying top level functions and then decomposes these functions into a hierarchy of sub-functions to form a functional architecture. The objective is to break down the system into simple tasks that can be performed by people, machines, and software that when combined with other sub-functions, will achieve the performance of the top level system functions (Chapman, 26).

Internal and external interfaces and functional attributes are also developed. An interface is the method that two functions use to communicate and exchange data elements between one another. An attribute, such as a criterion or an importance level, describes a function. Another important activity in functional analysis is ensuring each function is traced to a requirement from the earlier phase in the life cycle. This is beneficial because it assures that all requirements are met by one or more functions and if any of the requirements change, it is easy to identify functions that are affected. Although there are many tools and methods used to do functional analysis, the Functional Flow Block Diagram (FFBD) was used by our team due to its strength of performing functional decomposition and its ability to show functional relationships. A picture of an FFBD can be seen in Figure 2.

**Design Evaluation**

Although a functional model provides a strong learning tool and a visual representation of a complex system, the mere creation of the model would not be complete without a thorough evaluation of the design. This evaluation of the functional model exposes possible design flaws before any implementation begins.

There exist four major avenues to explore the strength of any system design. First, the commonality and standards criterion explores the standards held throughout the system and the overall scope of the system. The measurement of a system’s Commonality may include an evaluation of the minimum number of unique system elements. Not only should commonality investigate the hardware and software aspects of the system, but the criterion should also assess common elements amongst manufacturing processes and maintenance approaches.

Similar to exploring the Commonality of a system, surveying the structure’s overall complexity also proves helpful in evaluating the design. The goals of any system should include minimizing the number of mechanisms, unique hardware structures, and the number of functional overlaps. In addition to monitoring the growing complexity of a large system, a system designer should focus on the supportability issues the system creates. For this, one may look at the RAM, or reliability, availability and maintainability, of a large system design. By focusing on RAM early in the system engineering life-cycle, a system architect saves valuable time during system implementation.

Lastly, the modularity criterion of architecture analysis refers to the upgradeability of the system. Within this criterion, a good system minimizes orthogonality, or overlapping functionality, and ensures the system elements are fairly decoupled, or remain as separated as possible.

**METHODOLOGY**

**Requirements Management**

The IDOT PTC project has gone through the steps of RM mentioned previously. The customer, IDOT, provided the Systems Specification Requirement (SSR). This is the top level requirement document that states the customer needs. With the SSR in hand, the engineers at Lockheed Martin were able to break the system into five specific segments. The office segment, locomotive segment, communication segment, wayside interface unit segment, and the railway worker train segment.
segment make up IDOT-PTC. With each segment, more requirement documents were written to tailor to the specific needs of each segment. For example, the office segment has the Office Program System Requirement Specification (OPSRS) derived document. The document broke the Office Server into 22 distinct components. Each component with its requirements performed specific tasks. The development of these and further derived documents complete the requirement analysis stage of RM. In terms of requirement tracking and verification, the top level SSR were traced to lower level OPSRS. This step allowed our team to both refine the objectives and find extraneous or missing requirements.

**Functional Analysis**

Following Requirements Management, our team began Functional Analysis. The first step was to decompose each component and create Functional Flow Block Diagrams. This was the most tedious and laborious part of our project since each component could have over 50 functions and understanding how each component accomplished its tasks took quite a bit of time. Our team began by looking at all the requirements that traced to each component in an effort to get a general idea of what the top level functions would be. The top level functions for most components seemed to be a start-up, an operational, and a shut-down function. For the majority of components, the start-up and shut-down functions were very similar. Obviously, the operations that each component performed varied dramatically.

We looked at a variety of sources to help us determine what the operational functions would be. The IDOT PTC Concept of Operations helped explain, in greater detail, what some of the components did. From there we just looked at the SSR document. The SSR document is the parent of the OPSRS document in the hierarchical set of requirement documents for the PTC Program. Since the SSR document is higher up, it is more general and provides us a framework of the functionality of each component. Our most valuable resource was our original OPSRS requirements document. This provided some explanation in between requirements and broke down some of the operational tasks into more specific areas. After creating a function, it was traced to one or more requirements to maintain proper traceability. This process was extremely iterative and required many modifications.

In creating our functional architecture, our team used Functional Flow Block Diagrams (FFBD) in two different capacities. The first and most obvious was to see how the functions related to one another. The second way we used FFBDs was to actually create the functions. In CORE, one can directly add, change, or remove functions when viewing an FFBD. This allowed us to look at the requirements in a given section and create a set of functions that flowed logically together that satisfied the requirements. This method also enabled us to address areas of logic, such as recursive loops, “ANDs”, “ORs”, and “GO” “NO-GO”.

**Interfaces**

After our team completed the Functional Flow Block Diagrams for each component it was necessary to create interfaces between components. Our team went through the OPSRS requirements document and identified 92 places where one component talked to another. After properly entering these interfaces into CORE, we began to identify the individual functions that were responsible for the component-component communication and traced these functions to the established interface. Lastly, our team needed to identify the specific data elements that are transferred over each interface. Examples of data elements include a request, a report, or an advisory. Once all the data elements were entered into CORE, each element was traced to one or more interfaces.

**Design Evaluation**

Having developed a functional architecture for the IDOT PTC Office Server, our team developed a series of objectives to best evaluate the design. The first objective calls to check for omissions or unnecessary redundancies. In order to search for functional omissions, we analyzed the functionality described within the model and looked for breaks in the logical flow of the functions. Moreover, the team looked through the components to determine if their functionality overlapped or created unnecessary redundancy. The next objective requires the team to search for intuitive flaws within the logic of the design architecture. In order to complete this objective, the team searched the OPSRS document for missing or unstated requirements. Furthermore, the team investigated possible logic breakdown within the interfaces between components.

While the first two objectives focus on general design structure, the last two center on specific areas of the design. First, the team investigated the interfaces amongst components in order to determine the amount of coupling within the system. The team searched for common data elements and reported elements which interacted with multiple components. Finally, the team
tackled the error detection functionality of the system as a whole. Not only did we track the location of current error detection functionality within each component, we also focused on the reporting characteristics of these errors.

**CONCLUSIONS**

The goal of the Capstone project is to assist Lockheed Martin in its implementation of a Positive Train Control System for the Illinois Department of Transportation by providing valuable feedback to the team of design engineers. To achieve this goal, several steps of the systems engineering process needed to be performed: Completion of Requirements Management, Functional Analysis, and a Design Evaluation.

Requirement Management through requirement elicitation, requirement analysis, requirement tracking, and requirement verification was performed on the Office Server of the PTC System. The upper and lower level requirements were traced to show their relationships. The Office Program System Requirement Specification was broken down into 22 further components due to their specific tasks.

Functional Analysis for the Office Server of the PTC System was completed using several steps. After each requirement statement was traced to the appropriate component, each component was functionally decomposed. Functional Flow Block Diagrams were used to show proper sequence. Interfaces between components were defined and the appropriate functions were allocated to the interfaces. Also, data elements were defined and allocated to functions and interfaces.

After functional analysis was completed, our team was able to evaluate the resulting functional architecture and provide adequate feedback to the design team. The four major areas of the evaluation are:

- Omissions and Unnecessary Redundancies
- Inconsistencies in Logic or Requirements
- Component Interfaces and Coupling
- Error Detection

Following the completion of the three major steps of our project we were able to develop several deliverables to present to the Lockheed Martin design team. The deliverables for the project were:

- A report presenting the findings from the functional architecture evaluation
- User Manual for how to use CORE with respect to the IDOT PTC Project
- CORE database
- Website that contains all information related to the functional model
- A report on CORE’s simulation tool

**REFERENCES**


**BIOGRAPHIES**

Ryan P. Griswold, is a fourth-year Systems and Information Engineering major concentrating in Management Information Systems. Originally from Annandale, Virginia, he has accepted a position with First USA in Wilmington, Delaware.

Thomas Lai, is a fourth-year Systems and Information Engineering student from Burke, Virginia. He is still looking for a job and is considering attending graduate school.

Chris W. Pritcher, is a fourth-year Systems and Information Engineering student from Fairfax, Virginia. He has accepted a position with Capital One in Richmond.

Dong Shao, is a fourth-year Systems and Information Engineering student from Burke, Virginia. He will be yet pursuing another rigorous semester at the University of Virginia.