2015 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
SYSTEMS ENGINEERING (SE) TECHNICAL SESSION
AUGUST 4-6, 2015 – NOVI, MICHIGAN

Integrating P-Diagrams into Specifications

Kevin J. Dutcher
Maneuver Collaboration Center (mc2)
General Dynamics Land Systems
Sterling Heights, MI 48310

Many of the "ilities" (Reliability, Maintainability, etc) are afterthoughts in the creation of a specification, and are often relegated to a set of templated boilerplate requirements, that are largely ignored. The Reliability / Robust Design professionals often use a P-Diagram (Parameter Diagram) as a key part of understanding the system under design. A way of integrating the Reliability effort more into the mainstream of the design activity, and give them a stronger voice, is to put their P-Diagram right into the specification, before it gets released to industry. This paper describes the rationale and the manner in which to do this.

INTRODUCTION

Specifications tell stories, and most often, without the storyteller's presence. Without the storyteller present to provide background and context, to fill-in over-looked assumptions, to correct incorrect assumptions, and to accurately communicate that story, it is most likely that the story will not be read as intended and will not have a happy ending. In the world of contracting, misread specifications translates to projects that are over budget, over schedule, and don't meet the desired performance requirements.

For years, MIL-STD-490 (now cancelled) provided a template for Department of Defense (DoD) specifications, and most recently, MIL-STD-961 now serves that purpose. While both of these Standards called out key elements of the specification story (Performance, Physical Characteristics, Environmental Conditions, Workmanship, Safety, Verification, etc.), the standards provided no guidance on how to tell the story in a logical format, to weave the tale across the sections, to provide the storyteller (specification author) a way to ensure the story was complete.

A few years ago, written into a US Marine Corp RFP (Request for Proposal) was the requirement for a "Seamless Integrated Database" for the data on the program, spanning from Engineering to Production to Support.. That phrase, Seamless Integrated Database has stuck with me for 20 years. Who wouldn't want that? Across the years of developing specifications and searching for better ways to do it, I've come to develop a personal technique which provides structure to the story and thoroughness to the specification storytelling endeavor. This paper describes that technique, and then describes a way to improve it, by incorporating a Parameter Diagram (P-Diagram), being used in the Quality and Reliability worlds. This novel use of P-Diagrams allows the knowledge gained by the robust engineering activities during the concept phase, to be transferred to the Supplier, within the specification, and additionally, it helps to structure the specification itself.

First I must describe that personal technique starting-point (pre P-Diagram) that involves the use of a Context Diagram, which identifies the external interfaces, and relates all the performance requirements to those external interfaces. That technique will be described first, and then it will be shown how it can morph to include the P-Diagram, integrating directly the work being done by the Reliability personnel on the program.

While the recommendation to look at utilizing P-Diagrams within specification is meant to start the conversation within the industry on the topic, I do encourage you to at least, start using Context Diagrams in your specifications. So first, some background on Context Diagrams and how they should be used in developing specifications.

CONTEXT DIAGRAMS

By definition, a context diagram describes the boundary between the item under study and it's surrounding environment, the "context" where it will be deployed. Classical Systems Engineering trains this as a starting point - what is within scope for the Systems Under Study (SUD) and what is outside the SUD. There is a strong parallel with performing analysis on a thermodynamics problem; first, the boundary has to be set, then analyze what energy is going across the boundary. A similar diagram is the Boundary Diagram, used for

In the case of other Systems, besides pure thermodynamic problems, there may be energy going across the boundary, or matter, or signal. The knowledge that these are the three classifications of what may flow across the interfaces is really at the heart of trying to ensure completeness of the specification.

These external boundaries described by the Context Diagram are the exact same interfaces where the majority of the performance requirements are levied. Think of a System Under Study (SUD) with only two interfaces; one an input and the second an output. It appears exactly as a transfer function acting on the Input (Interface 1) and creating the Output (Interface 2). See Figure 1 for a diagrammatic representation of this.

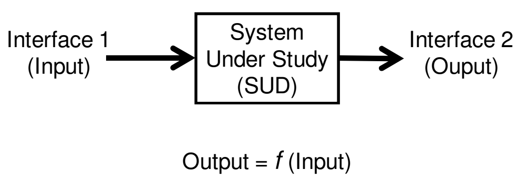

Figure 1. Simple System Correlates to a Transfer Function.

This transfer function is really the requirement of the SUD. Given the input at Interface 1, it must provide the Output at Interface 2. This connecting of the Interfaces to the Inputs/Outputs, and Transfer Functions (requirements) adds a structure to the requirements and the specification process that makes the specification inherently readable, logical, and complete.

A SUD with multiple interfaces (Inputs and Outputs) is only an extension of this simple problem, and keeping track of them all is essentially just a bookkeeping process. This is where the Context Diagram comes in.

Figure 2 below, shows a generic Context Diagram for a typical Ground Vehicle. A full page-width page graphic was included for readability, please refer to it on the next page.

One of the earliest documentations of Context Diagrams is Tom DeMarco's groundbreaking book "Structured Analysis and System Specification", published in 1978, which led to the Context Diagram being used in the software engineering industry.

MIL-STD-490 was initially released in 1968, a full decade prior to Tom Demarco's book. If the standard were to be written today, I assume that the Context Diagram would be a central part of it.

The Context Diagram appears to be similar to the DODAF High-Level Operational Concept Graphic (OV-1), but is meant as an Engineering document with data, acting as a reference and a defacto Table of Contents, and not just a cartoon. In the example in Figure 2, some of the External Interfaces are identified specifically, and some are categories, or collections of interfaces. For example, Interface B, Towing Vehicle, includes any vehicle that would be towing the System Under Study (SUD), whether it's an M88 or a Stryker.

The steps in creating the Context Diagram are iterative, and based upon the top-level system design. The core question is, what external elements does the system interface with? The next typical steps involve defining the function at that interface, and then further, what is flowing across that interface, and then finally detailing the physical interface. This view of Systems Engineering, from the point of view of the interfaces, provides a methodology to gauge completeness, by judging whether all the interfaces have been covered.

While this sounds like a fairly rigid process, it is, but there is also opportunity for innovation and creativity. In fact, it can be built right into the process. For example, a concept excursion can be examined where the vehicle doesn't interface with Towing Vehicle. What type of concept may come out of this? Perhaps a disposable, single-use vehicle that doesn't ever get recovered if damaged, which may lead to a very inexpensive and light vehicle.

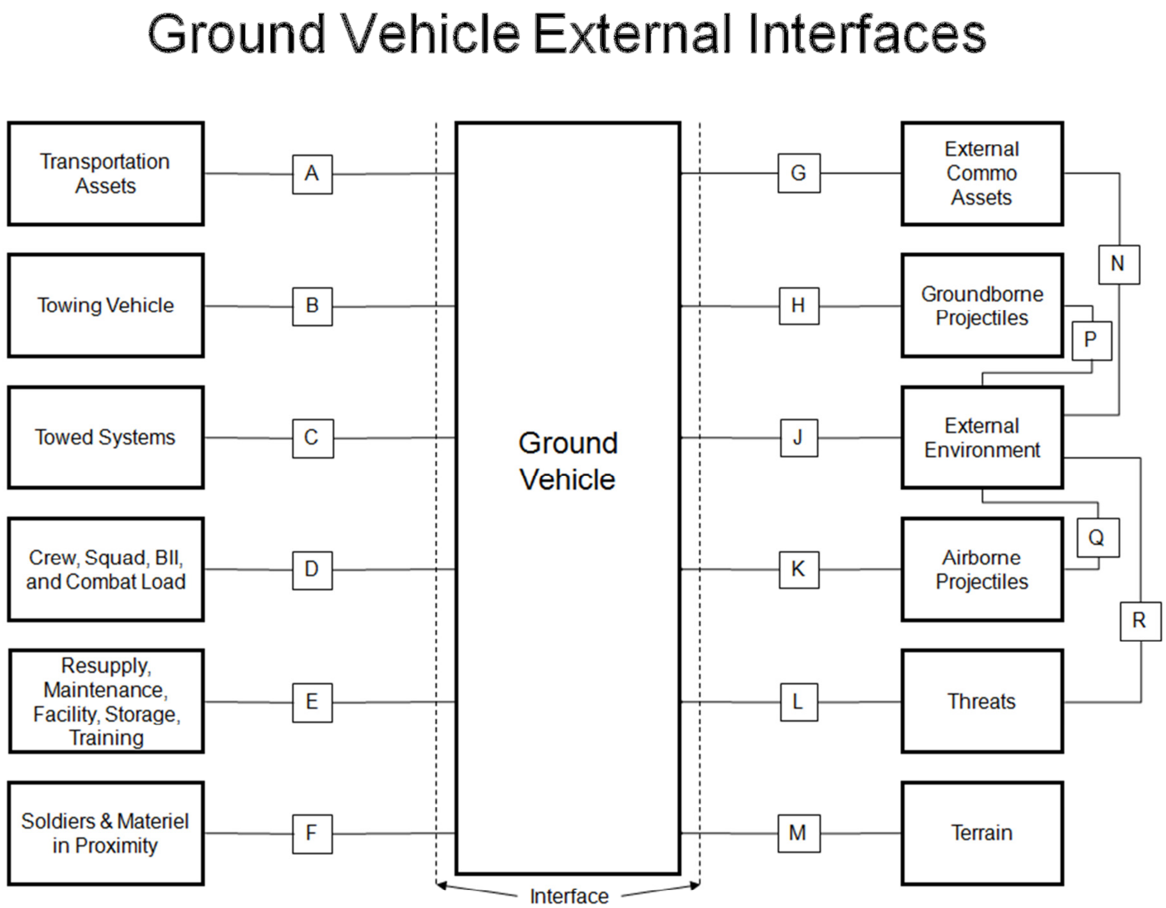

Figure 2. Context Diagram for a Generic Combat Vehicle

The Context Diagram is introduced in the introductory section of the Specification, typically within Section 3.1, and essentially creates a Table of Contents for the requirements in the Performance section, which can now be ordered by each of the external interfaces. For example, the headings in this section could read:

3.2.1 Performance
3.2.1.1 Transportation Assets
3.2.1.2 Towing Vehicle
3.2.1.3 Towed Systems
3.2.1.4 Crew, Squad, BII, & Combat Load
etc

Under each of the headings, for each of the interfaces, the performance requirements would be placed, list the functional requirements, the performance parameters, and the conditions.

This kind of structure within the Performance section of the specification increases readability and helps to ensure that all the performance requirements are covered. I have used this approach for personal efforts many times, and have used it in two formally released specifications where it was well received by the Suppliers who received and responded to the Request for Proposal to which it was attached.

So with the Context Diagram in the specification as a starting point, I have now been introduced to the Parameter Diagram (P-Diagram) by Reliability Engineers. Having been exposed to many of the Quality Tools that came out of Japan, I was interested in it, and it appeared it was firmly ingrained to how the Reliability Group was doing their work, going forward. There is a Chinese Proverb that states: If a tiger enters the temple make it part of the ceremony. The point of the saying is that if you can't change something, embrace it. That is was happened to me with the Context Diagram and the P-Diagram. I had made no headway trying to formally introduce Context Diagrams into all the specifications, but thought if I moved from Context Diagrams, to the P-Diagram, which could be modified to capture all the information in the Context Diagram, there may be enough critical mass to get it into all of our specifications.

The Parameter Diagram (P-Diagram)

Now some background on the Parameter Diagram (P-Diagram). The intent of the P-Diagram is to fully understand the relationship between the Inputs, Ideal Response (Output), Noise Factors, and Control Factors. It's basis was trying to increase signal/noise ratios.

Figure 3 shows a typical P-diagram. It looks a lot like a Context Diagram – the System in the middle and External Interfaces around the outside. The P-Diagram is more interested in the functions coming across the interfaces than the external interface elements themselves, because by the time the typical P-Diagram is developed, all the decisions and trade studies, on what the external interfaces are, have already been made. Adding the identification of the External Interfaces to the P-Diagram would allow these two models to merge.

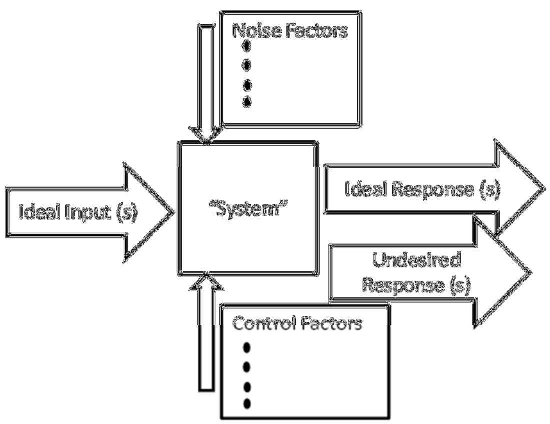

Figure 3. Standard P-Diagram.

The P-diagram is an organized, formal way to approach the design problem. To learn more about the P-Diagram in use, go to: http://thequalityportal.com/p_diagram.htm or http://www.systems2win.com/solutions/P-Diagram.htm

Each of the elements identified on the P-Diagram are, by definition, external to the System under study, and would, by definition, show up on a Context Diagram. They have tremendous overlap.

Sharing Knowledge

As the System Designers are architecting the system, and allocating functions and performance to lower level components, as documented in the specifications, part of this activity is likely delving into areas of the P-Diagram process, with questions such as; where do I make the interfaces so they are most robust to variation, how do I de-sensitize the system to environmental extremes, etc. As this information is gained, and decisions made, the most actionable place to put the information is in the specification. Let the materiel developer know what the "noise factors" are, that you have decided to loosen the tolerances on this one interface due to cost. All this information should be captured in the specification as the developer is the one that must take it into account.

Telling the Story

So let's look at how this might work, by concentrating on a single input, output, and transfer function, based on the Towing Vehicle external interface and the resulting friction relative to the terrain, Figure 4.

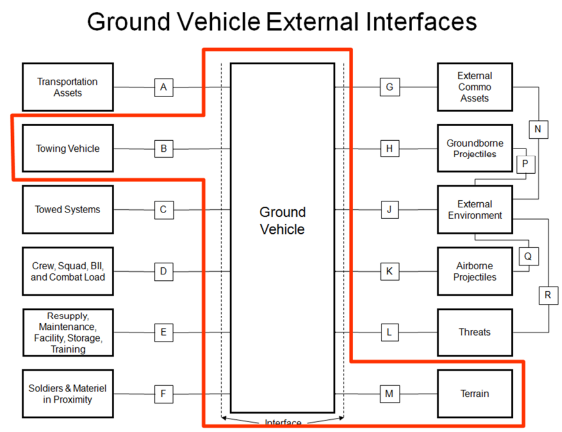

Figure 4. Context Diagram for a Generic Combat Vehicle

Detailing only those two external interfaces, and combining them with the P-Diagram gives Figure 5 below. A mashup of the two diagrams. A top-level function has been added across the two external interfaces, F.1 Non Self-propelled Movement. This function would have a number of performance parameters associated with it, such as the pulling force required, etc.

This diagram looks very useful. As if it could become the home of best practice and lessons learned in that area, used as a template, a Knowledge Management storehouse for things that the design engineer needs to look out for, prior to the design effort starting. Useful both for institutional knowledge and for the specification on that specific program. The right information transmitted at the right time. Isn't that what a specification is supposed to do?

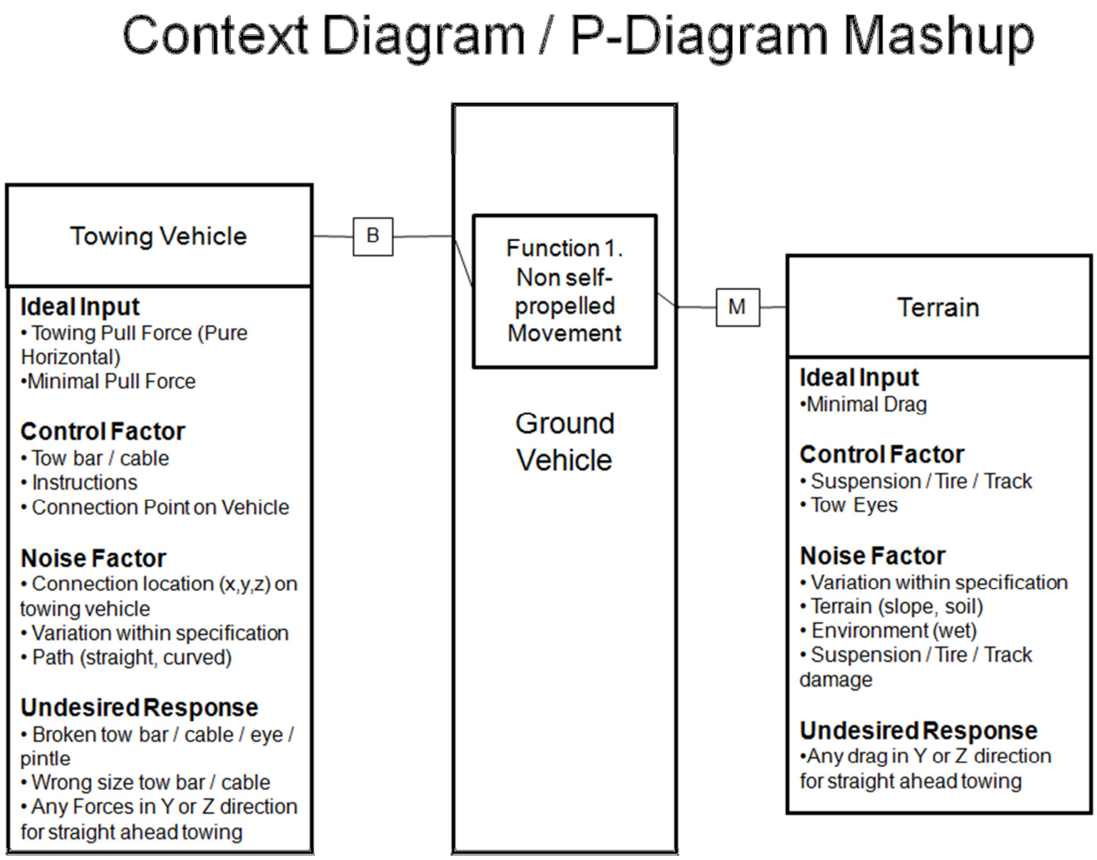

Figure 5. Context Diagram / P-Diagram Mashup

So now imagine that the specification story can be told like this.

3.2.1 Performance
3.2.1.1 Function 1
3.2.1.1.1 Ideal Response 1
3.2.1.1.2 Noise Factors
3.2.1.1.3 Control Factors
3.2.1.1.4 Undesired Responses

3.2.1.2 Function 2
etc.

This format allows the System Integrator to pass onto their supplier, more of the information that they know, relative to the System design, and starts the Supplier thinking in a detailed Robust Engineering manner from Day 1. It almost directly lays out in words what would comprise an optimization equation to use during the SUD design effort. It also lends itself directly to developing the FMEA on the SUD, as it contain the Functions, the desired output (Ideal Response), the Noise Factors, the Control Factors, and Undesired Responses.

Summary

This paper is not an attempt to convince organizations to start using P-Diagrams in their design process, but to convince organizations that, if you are already using P-Diagrams, consider doing them earlier, and including the diagrams in the specifications, linked to a context diagram. As any information that you know about the System Under Study, should be passed onto the system developer, as they are the ones that need the information. They are the ones that are listening to the story that the specification is trying to tell. And as the saying goes, “you don’t get what you don’t specify”.

Approved for public release, LogNo. 2015-33, Distribution Unlimited, 06/07/2015.