THE C+C DSM METHOD: MANAGING CHANGE PROPAGATION WHILE
PROTOTYPING A ROBOTIC INSPECTION SYSTEM
FOR DRY NUCLEAR WASTE STORAGE CASKS

A Thesis in
Mechanical Engineering
by
Jennifer E. Bracken

© 2018 Jennifer E. Bracken

Submitted in Partial Fulfillment
of the Requirements
for the Degree of
Master of Science

May 2018
The thesis of Jennifer Bracken was reviewed and approved* by the following:

Sean Brennan  
Professor of Mechanical Engineering  
Thesis Advisor

Timothy W. Simpson  
Paul Morrow Professor of Engineering Design and Manufacturing

Karen Thole  
Professor of Mechanical Engineering  
Head of the Department of Mechanical and Nuclear Engineering

* Signatures are on file in the Graduate School.
ABSTRACT

In prototyping complex systems, concepts often reach a point where incremental modifications are insufficient, and a high-level redesign is required. This phenomenon can be time-consuming and expensive, particularly when physical prototypes are involved. As time and money are frequently the limiting factors for systems engineering prototyping projects, engineers wish to avoid significant design overhauls in the middle of a product’s development cycle. However, because it is difficult for engineers to anticipate project-specific design issues leading to high-level redesigns, and because high-level redesigns can result from seemingly minor changes in components, avoidance of unexpected and catastrophic design change can be challenging. This thesis categorizes these changes and explores new methods for approaching product design and prototyping that avoid major mid-development redesigns of a product.

In order to identify the impact of prototype changes, this thesis employs Design Structure Matrices (DSMs) to examine the changes in an on-going real-world design project involving repeated design, fabrication, and testing of physical prototypes. Changes to the prototypes during the development cycle are analyzed in DSM form before and after changes were made to the system. These are presented to show change propagation to other component connections. These DSMs are completed using both a standard method and a proposed new method, called the C+C DSM Method. This new method includes constraints and components in one matrix, with the goal of eliminating the need to examine secondary and tertiary connections to predict change propagation. Compared to conventional DSM methods, the C+C DSM method is intended to offer a fast and flexible method specifically for an evolving prototype project, where time can be
a limiting factor. By presenting this method, this thesis intends to provide a decision-making tool for prototyping so that implementation of a change can be made with full awareness of its impact.

The project examples motivating this thesis work are taken from a systems design prototyping project that involved several universities and industry partners, including Penn State, Michigan State, University of South Carolina, the US Department of Energy, and Holtec International. This project involved the design and prototyping of a robot to inspect nuclear dry waste storage casks. This robot carries experimental sensors through vents and up and down the sides of the storage casks. The robot and its delivery system went through a number of different prototype iterations, both as individual subsystems and as a final system. The prototype iterations discussed in this thesis were chosen for analysis because they were thought to be minor at the concept state and in the analysis of conventional DSMs, but resulted in major design changes once implemented. Had these major changes been identified earlier, the design process would have been far more efficient. The results show that the C+C DSM method developed in this thesis could have predicted the large number of “knock-on” changes associated with each change that spurred an iteration.
# TABLE OF CONTENTS

List of Figures ........................................................................................................ iii
List of Tables ........................................................................................................ iv
Acknowledgements ............................................................................................... v
Chapter 1 Introduction ......................................................................................... 1
  1.1 Motivation ..................................................................................................... 2
  1.2 Goals ............................................................................................................ 3
  1.3 Outline of Remaining Chapters .................................................................... 5
Chapter 2 Review of the Literature ..................................................................... 6
Chapter 3 Case Study Project Background .......................................................... 12
  3.1 Inspection of Dry Nuclear Waste Storage Casks ......................................... 12
  3.2 The NEUP Cask Inspection Robot Project ................................................. 13
Chapter 4 Illustrating the Proposed Change Propagation Management Method via Key Examples ..................................................................................................................... 19
  4.1 Overview of the Insertion Arm Subsystem .................................................. 19
  4.2 Key Examples of Changes that Spurred Many Knock-On Changes .............. 21
  4.3 Overview of the New C+C DSM Method .................................................... 27
  4.4 First Example of the C+C DSM Method ..................................................... 28
  4.5 Second Example of the C+C DSM Method ................................................ 30
Chapter 5 Thesis Outcomes and Future Work .................................................... 42
Bibliography ......................................................................................................... 45
LIST OF FIGURES

Figure 1: A basic example DSM for 8 interconnected parts .............................................. 6

Figure 2: Sensor train of robotic inspection system prototype atop an empty storage cask with the storage cask lid removed. The sensor train is about to perform a mock inspection on the cask at the Holtec facility in Pittsburgh, Pennsylvania. .................................................. 15

Figure 3: Two people use a high lift to inspect the entrance to ventilation shafts on one of the dry nuclear waste storage cylinders ......................................................... 16

Figure 4: The PRINSE robotic inspection system attached to a dry storage cask at Holtec in Pittsburgh, PA. ................................................................. 17

Figure 5: The sensor delivery subsystem "arm" inside an empty waste storage cask at Holtec in Pittsburgh, Pennsylvania, with the hump and garage explicitly labeled. ...................... 18

Figure 6: The insertion arm segment of the robotic inspection system, specifically the garage segment. The image on the left shows the garage closed, and the image on the right shows the garage open to display the cars housing the sensors. The cream-colored component at the center top of the left image is the hump. The center-bottom aluminum component in the left image is the garage lid. ......................................................... 21

Figure 7: The spring reel mounted to the garage lid, with a cable attached to the robot ........ 23

Figure 8: DSM of the garage portion of the insertion arm before the addition of the spring reel. 24

Figure 9: DSM of the garage portion of the insertion arm after the addition of the spring reel. Green boxes with represent connections added upon implementation of the spring reel, orange boxes with represents connections where problems occurred due to the addition of the spring reel ......................................................... 24

Figure 10: The same DSM as Figure 9, but with unexpected changes boxed for emphasis .... 26

Figure 11: C+C DSM of the garage portion of the insertion arm before the addition of the spring reel. The constraints are now included in the matrix in a manner identical to components, as per the new method for tracking design change propagation. The interactions between Components 14 and 15 and the rest of the system are boxed .................. 29

Figure 12: Conventional DSM of the garage portion of the insertion arm before the addition of the spring reel, without constraints. The connections for Components 14 and 15 are boxed, and can be seen to not include components 11 or 12 ........................................ 29

Figure 13: C+C DSM of the garage portion of the insertion arm after the addition of the spring reel. Green boxes, denoted “o” for origin of a new connection, represents connections added upon implementation of the spring reel, orange, denoted “p” for possible problem, represents connections where changes occurred due to the addition of the spring reel. Connections of Components 14 and 15 are boxed. The constraints are all included as per the new C+C method ................................................................. 30
Figure 14: Conventional DSM of the system before changing to the clamshell garage lid......32

Figure 15: C+C DSM from before the clamshell garage lid change occurred, including constraints in the C+C style.................................................................32

Figure 16: C+C DSM with only physical connections, prior to the garage lid modification to a clamshell design. Connections of the component that was purposefully changed to create the clamshell design are boxed. .................................................................34

Figure 17: DSM of the insertion arm system after the clamshell garage lid modification. Connections that became problematic are orange with the letter “p” for possible problem, those that were added are green with the letter “o” for origin of connection. The connections that were modified on purpose are boxed.................................................................34

Figure 18: C+C DSM created to show connections in the system using the C+C method, before the change to the clamshell garage lid design. Red boxes are around the constraint connections affected by the change to a clamshell style garage lid. Component 11 was the only component intentionally modified, and it is boxed in orange...........................35

Figure 19: C+C DSM of the system after the garage lid was modified to be a clamshell design. For clarity, the constraints are boxed, and arrows show the component connections that the constraint modifications might affect. This is only done in the y direction to make the image simple to view, and due to symmetry it shows all affected component connections......38

Figure 20: DSM created for the insertion arm before the addition of the spring reel. This DSM has been sorted using a MATLAB algorithm to form two modules, which are shown with the red boxes.................................................................40
ACKNOWLEDGEMENTS

I would like to thank the following people for making this thesis possible:

Dr. Sean Brennan, for all of his advice, support, and guidance.

My parents and family, for their unconditional love and support.

The PRINSE team, for helping to make the project a success.

Dr. Timothy W. Simpson, for his assistance.

This material is based upon work supported by the Nuclear Energy Universities Program under award number DE-NE0008266.
Chapter 1

Introduction

During the prototyping of complex systems projects, many incremental changes can occur over time. As engineers are making changes, they often reach a point where one seemingly small change in one component results in a sudden need to make a large number of changes in other, seemingly unrelated components. In more extreme cases, one incremental change made by one party can necessitate the redesign and rebuild by many other parties of many components of their respective subsystems. This cascade-of-changes phenomenon can cause delays, high additional costs, and general difficulty for the team as a whole. This thesis explores a new method for identification of changes with potential to cause a cascade of problems in that manner. It applies that method to a complex systems design project with an emphasis on physical prototyping. This project, design of a robotic inspection system for dry nuclear waste storage casks, went through a large number of prototype iterations, providing a relevant, real-world example to show the efficacy of the proposed approach. Many of these design iterations resulted from seemingly minor design decisions that cascaded to become large-scale design changes. The repeated observation of this phenomenon seeded the ideas presented in this thesis.
1.1 Motivation

While the resulting problems and costs of making one incremental change to a component can be large, the process of design for single components – material selection, FEA analysis, pricing analysis, etc. – is well understood and part of the common experience of design engineers. However, when seemingly minor changes to one part rapidly cascade into required changes elsewhere in a complex system, this situation is not easy to predict with current design methods. Gantt charts and other techniques create schedules and predict delays based upon task completion by individuals or teams. However, these methods do not address the specific engineering decisions that can cause an avalanche of changes to occur within an engineering design. There is an engineering need for the development of design analysis tools, specifically ones that can predict when a decision to make a change will necessitate a cascade of other design changes often in seemingly unrelated parts within a complex system.

When such an “avalanche of change” occurs, the design teams involved often find themselves organizationally or even geographically separated, further challenging decision-making on how to address these issues. This reveals another weakness of current, commonly-used design methods: the need to address what communication is necessary between team members or sub-teams in order to avoid problematic changes originating at subsystem boundaries and interfaces. While it may appear ideal for all parties to simply employ a “talk more” strategy, a focused and strategic communication plan reduces time and money spent on communication. Such a plan would increase time available for technical work, which would boost overall productivity. Even in direct costs, the cost of a meeting, for example, increases
with the number of attendees due to overhead charges. Therefore, determining which parts of which teams will interact becomes not only important from a design standpoint, but also from a project-funding standpoint.

A method to both predict when a design change would create an avalanche of changes, and how different parties should be communicating to avoid such design changes is desirable. Current methods in project planning and design do exist and are reviewed in the remainder of the thesis. However, when conventional methods were applied to the case study for this thesis, the implementation examples show that there were shortcomings. This observation guided the creation of a new design analysis method for tracking and predicting change propagation in systems design projects, one that is both simple to implement and inclusive of both component connections and project-specific constraints. This new method, and the examples that motivated its development, are the subject of this thesis.

1.2 Goals

The specific goal of this thesis is to develop a new method, the component-and-constraint-based C+C DSM method, to help manage change propagation during prototyping. The thesis demonstrates the efficacy of this method by applying it to a case study of prototyping a robot for inspection of nuclear waste casks. This method is a new application of a technique used to look at connections between people, teams, and components: Design Structure Matrices (DSMs).

The conventional DSM technique, and variants of it, are often used by the design community to determine how to divide large projects into sub-teams and to show where those
sub-teams need to interact. Additionally, it is used regularly within the literature to visually represent physical, spatial, and functional connections between components. While a DSM naturally facilitates insights that help achieve these goals, current DSM methods require completing multiple complex matrices, or completing matrices that include multiple connection styles, in order to visualize team interaction based upon components, and team interaction based upon constraints. This thesis presents a streamlined method that merges all of these applications into one modified matrix, with the intent of giving users an overview of the prototype system and possible changes with minimal time investment. This yields one holistic method for managing change propagation in systems design prototyping projects. The C+C DSM method’s emphasis on rapid, intuitive implementation is designed to suit the needs associated with prototyping: flexibility and speed.

The proposed C+C DSM modifies more traditional DSMs by using one large matrix to better show whether a seemingly incremental change might become an extreme change and thus propagate changes to other component connections. Incremental changes are hereafter defined as small physical modifications to one or a small number of components that change the system design slightly and improve its overall performance. This type of change, by definition, does not spark a cascade of other changes. In contrast, extreme changes are defined as those changes that spur a multitude of other changes in their wake among many components in the system. Unfortunately, extreme changes can also involve minor physical modifications to one or a small number of components, and thus may appear on the surface to be incremental in nature. Delineating which design changes are incremental and which are extreme is a fundamental
objective of using DSM matrices, and thus, the goal in the C+C DSM method is to further increase the decision-making party’s awareness of consequences during prototyping, leading to better outcomes.

1.3 Outline of Remaining Chapters

This thesis works to develop the C+C DSM method to help manage change propagation during prototyping and demonstrates the efficacy of this method by applying it to a case study of prototyping a nuclear waste storage cask inspection robot. In order to give context to this technique, Chapter 2 of this thesis begins with a review of current literature related to DSMs and change propagation. DSMs can determine how to divide large projects effectively among sub-teams, and where those sub-teams will need to interact. They can also visually represent physical, spatial, and functional connections between components. Examination of these previous applications identifies areas for growth with the new method.

Chapter 3 of this thesis then covers the background of the project used as a case study, namely, design of a robotic inspection system for dry nuclear waste storage casks.

Chapter 4 explains where this project provided key examples of change propagation that show the efficacy of the proposed method. This chapter also presents an overview of the proposed DSM method. This method is a novel application of a technique used to look at connections between people, teams, and components: Design Structure Matrices (DSM). This chapter specifically shows two examples of extreme changes using both a classic DSM method and the new C+C DSM method. Both examples are from prototyping the insertion arm of the inspection robot.

In Chapter 5, after examining those examples, this thesis concludes with an analysis of the results and suggestions for future work.
Chapter 2

Review of the Literature

The design analysis techniques outlined in this thesis are a unique application of the component-based Design Structure Matrix (DSM) methodology, which is an analysis technique used extensively to show connections between parts of a system [1]. The component-based style of DSM involves listing the components of a system as the title of each column, and as well as the title of each row. Connections between the components are represented by placing marks in the box where the two components’ rows and columns intersect; see Figure 1 for an example.

The connections between components are typically spatial, energy, information, or material [1].

![DSM Example](image)

Figure 1: A basic example DSM for 8 interconnected parts.

The DSM in this form allows engineers to visually represent how systems are interconnected, which can in turn inform what part-to-part connections to monitor if design changes are needed. The visual analysis of product complexity is a common application of DSM, as the literature can attest [1]–[10].
In the literature, DSMs are useful to specifically display how changes to any part of the system might ripple into changes to the rest of the system, a phenomenon called change propagation. Examples of using DSM change propagation techniques include applications to complex physical systems such as helicopters, engines, and XEROX printers [7], [9], [10], with specific techniques such as Change Prediction Methods (CPM) [8] and Function-Behavior-Structure (FBS) linkage models [10]. In fact, Hamraz and colleagues, in their work to develop a requirements-based engineering change management method, identify 54 unique techniques for change management as of 2013, because management of engineering change is such a prevalent challenge in systems engineering projects [11]. With such extensive options for managing engineering changes, there are methods for classifying nearly every possible situation. DSM change propagation methods include many nuances, such as components in matrices being classified as propagation absorbers or propagation multipliers, based upon the probability they will need to be changed when alterations are made to other components [12]. The manner in which the changes propagate has been classified as well, into “avalanches”, “ripples”, and “blossoms” [7]. Techniques using DSMs even show how changes affect systems overall, as “technology infusion effort” related to specific changes [9]. One particular type of change this thesis will explore is the “knock-on” change; knock-on changes are defined as those that occur because another change was made [10]. These knock-on changes are further defined as changes that are made not on purpose, but as a reaction to design problems created by making another change first.

While prior work extensively examines change propagation of complex systems, DSMs are also used regularly to make decisions about how to schedule the project. An example of this is one DSM technique similar to a Gantt chart for project management [3] by Minogue. His
technique combines ordering tasks in a manner similar to a Gantt chart with tracking changes in the matrix as small feedback loops to show how changes will affect the system schedule. This technique is helpful for planning schedules that can take into account engineering delays which can cause large increases in cost for projects, and delays in bringing new products to market [10], [13]. This prior work is relevant to later discussions, as this thesis also includes using DSMs to determine which parts of the system are likely to have high time costs to redesign.

Another design technique similar to a DSM that helps users to make design choices is the House of Quality [14]. This technique allows users to weigh the relative importance of different facets of a design. The roof of the House of Quality is similar to a DSM in that it shows relations between components or engineering characteristics with a mark indicating whether two items affect each other. This technique relies on the users’ knowledge of the project to determine these weights and to reach conclusions about which parts of the system are most important in order to obtain a design result. The information included in the House of Quality that is often lacking in DSMs includes: customer attributes, engineering characteristics, and target values. These features make House of Quality matrices valuable for setting targets and seeing how the customer’s desires interact with the engineering requirements, but they are not typically used to evaluate designs at an individual component level.

DSM techniques can also be applied to people and teams, particularly to show how project sub-teams or parties should be communicating. For example, Sosa et al. discussed how DSMs could be used to illustrate whether the appropriate people are communicating on large systems design projects [6]; this same discussion was the topic of an article and previous work by McCord et al. [5], [6]. Their findings along with others show that clustering DSMs can reveal how to divide projects into modules [15] or separate tasks among parties [16]; their suggestions
were to sort the matrices so that the connections form clusters, and to divide the project based upon the groupings that appear. This sorting can be accomplished manually, or by using various clustering algorithms that seek to move part groupings, or clusters, into connections close to the diagonal. The clusters delineate groups to create sub-teams or product modules, with outlying blocks or connections representing team interdependencies where these sub-teams need to communicate, as well as where the product will require module interfaces [13].

These prior studies deeply examine the use of the DSM techniques with a component-centric focus. This thesis builds upon this body of work by expanding the DSMs to include details on how the components relate to functional constraints, and to flexibly add and remove constraints with understanding of their impacts on the product. The prototyping project that occurred during this thesis experienced some changes similar to the “hidden dependencies” referenced by Hamraz and colleagues, but these dependencies were approached slightly differently in this thesis than in their FBS linkage model [10]. This led to the identification of the final need for the new DSM method: it had to account for not only components, but also their relations to constraints imposed by the implementation environment.

To contrast the work in this thesis to prior literature, this thesis seeks to combine five features into one simple-to-implement DSM-based method for prototyping:

1. DSMs predict which parts have potential to be affected by changes via looking at physical connections.
2. DSMs assist in determining which parts of the system are likely to have high time costs to redesign.
3. DSMs show how the project can be divided among involved parties.
4. DSMs show where different team groupings of the project must communicate.

5. DSMs can account for not only components, but for their relations to constraints imposed by the implementation environment.

This thesis seeks to formulate all of these features in an intuitive, flexible, and rapidly implementable method for managing change propagation during prototyping.

The extensive body of work referenced earlier has made predicting and managing change propagation far easier than it was in the past [11], yet much of DSM usage still requires extensive time and expertise to create the matrices and include the necessary types of connection styles, rankings, and/or values. This complexity can cause existing tools to inadvertently exclude inexperienced designers, as they may not be able to properly use them [17]. The proposed new method seeks to avoid this problem by being simple to use, and requiring little prior design experience.

This new C+C DSM method seeks to remain true to classic DSM principles of simplicity, in that that a single informational structure – a matrix – can show how components are connected, what system components will be affected by changes, and what changes may require significant time and effort due to “knock-on” changes. It also seeks to maintain the ability to show how the project can be divided, and where different parts of the project need to communicate. Compared to conventional DSMs, the C+C DSMs are novel in that they combine components with constraints, as shown in Chapter 4. In most other ways, this thesis implements existing features of DSM techniques. Overall, this thesis intends to present an option that is very similar to many existing methods; one that combines the features useful for the prototyping project undertaken during this thesis but is faster and easier to implement, and one that does not require significant re-training if DSM methods are already well understood.
In the next chapter, Chapter 3, the specific project motivating these new DSM methods is discussed in further detail. This project involved prototyping a complex system, which made it a useful case study for testing the method proposed in this thesis to manage changes during prototyping.
3.1 Inspection of Dry Nuclear Waste Storage Casks

This project involved the design and prototyping of an inspection robot for nuclear waste storage cask inspection [18]. This project was undertaken because the United States has six decades of experience in the development and commercial usage of nuclear energy technologies, and this usage has resulted in spent nuclear fuel which must be safely stored. While there are significant security and cost concerns to reprocess spent fuel, the energy potential of such “waste” is so advantageous that future access may be necessary. Specifically, reprocessing or use of spent nuclear fuel in breeder reactors offers society an almost inexhaustible source of carbon-free power; this is possible only as long as spent fuel is not entombed in an irrecoverable manner [19].

To date, the typical practice is to store spent fuel in cooling pools located nearby the source reactors, and then move it to dry-storage casks typically located near nuclear plants. Until recently, the long-term storage plans for spent fuel consisted of moving the dry-storage casks to the Yucca Mountain facility in Nevada, but this was recently shuttered [20]. Thus, dry storage casks – some of which are nearing their 40-year certification life – are presently the long-term storage solution for spent nuclear fuel and waste. These casks are typically located outdoors in
secured grounds nearby operational nuclear reactors, within vented concrete structures called “overpacks” that provide physical protection and radiation shielding to the stainless steel storage casks.

This extended use of dry storage for spent nuclear fuel from U.S. nuclear power plants makes it desirable to confirm the structural integrity of dry storage casks through on-site, non-destructive inspections. The Nuclear Regulatory Commission (NRC) requires independent spent fuel storage installations (ISFSIs) to submit an aging management program (AMP) as part of the re-licensure process [21].

3.2 The NEUP Cask Inspection Robot Project

The project referenced during this thesis sought to utilize robotic delivery systems that carry non-destructive sensors to allow remote inspection of the aforementioned casks for an AMP. The novel technology required for this robotic inspection system was designed by parties located in various departments at four different universities, and as such presented opportunities to apply engineering change management techniques. The name of this thesis’ robotic inspection system was the Proactive Robotic Inspection of Nuclear Storage Enclosures, or PRINSE.

The inspection process developed as part of this thesis work could be used for many applications, but it was focused in particular on measurement of Stress Corrosion Cracking (SCC), a nuclear dry-storage cask degradation process that could occur in stainless steel after long-term outdoor storage in an environment that has high levels of salts in the air. For SCC to occur, it requires the simultaneous presence of three factors: (1) a susceptible material, (2) a driving force, and (3) a corrosive environment. Material susceptibility could occur in practice
because austenitic stainless steel, the primary material of dry-storage casks, can have grain boundaries that might become chromium-depleted when the units are welded. The driving corrosion force also originates with welding, which often results in high thermal residual tensile stresses that may initiate and drive crack growth. Finally, the environment can drive corrosion in cask storage systems; airborne chloride salts are generally present in coastal environments where many casks are stored, and these salts could deliquesce on the cask surface and thus cause a low-concentration salt coating.

The inspection system analyzed later in this thesis houses unique sensing systems. These sensors seek to measure whether the surface environment of the cask enables corrosive activity and assess the presence of cracks in the stainless steel cask walls. Surface chloride sensors include laser induced breakdown spectroscopy (LIBS); field tests showed that this LIBS sensor can provide an in-situ characterization of the surface presence or absence of chlorides [22], which would indicate conditions for SCC to occur. The measurement of potential cracks in the stainless steel is achieved by utilizing an Electro-Magnetic Acoustic Transducer (EMAT) unit developed by one of the sub-teams specifically for nuclear cask inspections. Both the EMAT and LIBS technologies are newly deployed concepts that were only recently demonstrated at Penn State in October of 2017.

A key challenge in the inspection process is that inspection sensors need to be delivered into high temperature and radiation environments in confined spaces. Thus, an integral design focus is the geometric optimization of the robot and sensors for a particular cask and the careful measurement and control of the robotic position during inspection. These constraints had to be accounted for in every design iteration of the prototype system, and they strongly affected design choices made during the project.
In order to verify various prototypes of the system were functional, the system design and demonstrations utilized specific HI-STORM 100S-series models of the Holtec dry storage cask. For testing, a 1:1 scale mockup was constructed. Additionally, the full robotic inspection system was tested on an empty dry storage cask at the Holtec facility in Pittsburgh, Pennsylvania, pictured in Figure 2.

The overpack systems were designed to protect and shield the dry-storage casks, which meant that access to the casks by the robot was limited to small vents near the top of the cask. These vents are dozens of feet above the ground. Figure 3 shows people using a high lift to inspect the vent entrance of a cask.
As a result, a limiting design requirement was that any inspection components that access the cask must fit through the vent and be easy to transport by one or two people who are likely operating within a high-lift or similar equipment. Also, where possible, key equipment must be protected from heat and radiation.

To achieve these design requirements, the core actuation and control systems were mounted outside the cask using a unique and robust vent mounting system; the technologies to secure the robotic system within the cask for robust retrieval were developed as part of this project. This mounting system is pictured in Figure 4, attached to a storage cask at the Holtec facility that has been lowered into a pit for easy access.
Successfully managing all of the design constraints associated with this project allowed it to grow from concept stage to a Technology Readiness Level of 5 over the course of three years.

As illustrated later in this thesis, the vent geometry constraints are very specific to particular models of overpack and cask combinations. Each cask imposes specific but well-defined geometric constraints on core components used in the inspection. These constraints had to be included in the DSM techniques in order to fully represent all of the considerations for the prototype.

The more elaborate and complex prototypes near the end of the three-year project involved integrating components and constraints among many sub-teams, and any design changes had to be carefully considered to avoid extensive “knock on” changes to other
subsystems, or in some cases to their own subsystem, due to time and budget constraints. Within the sensor delivery sub-group alone, there were five different subprojects worked on by different parties. The design challenges and potential solutions created by the complexity of these systems are discussed in extensive detail in Chapter 5, with particular attention to the hump and garage. These components can be seen in Figure 5.

Figure 5: The sensor delivery subsystem "arm" inside an empty waste storage cask at Holtec International in Pittsburgh, Pennsylvania, with the hump and garage explicitly labeled.
Chapter 4

Illustrating the Proposed Change Propagation Management Method via Key Examples

4.1 Overview of the Insertion Arm Subsystem

The robotic inspection system required the integrated design of dozens of subsystems and components, and this design process revealed repeated situations where incremental design changes of one part or subsystem caused cascading and sometimes major changes in other components or subsystems. To illustrate this effect, and to show the use of methods to discover this aspect in the design process, this thesis presents examples from the PRINSE NEUP project, a project that experienced this phenomenon repeatedly. For the sake of simplicity, the following examples focus solely on the prototyping of the insertion arm segment of the inspection system, and particularly on the design of the so-called “garage” area of the robot.

The so-called “garage area” of the insertion arm holds the wheeled cars that house the sensors during insertion into the vent. In this way, it mimics a garage for larger vehicles, which is the origin of its name. This garage has two key functions. The first is to house a mechanism for initiating the cars’ motion up and over the hump at the end of the arm, so that the cars can enter the vertical gap area surrounding a cask, and a cask inspection can begin. The second purpose of the garage is to prevent cars from falling off the arm during vent insertion.

The garage mechanism went through many iterations to reach a successful final design. Two of these changes necessitated a particularly large number of unexpected design iterations, and because these design changes were well documented and concentrated into a small number
of interacting components of the overall system, the garage design changes provide the examples discussed later in this chapter.

While the garage holds the cars securely during inspections, it also makes access to the cars easy during testing by using a clamshell design for the roof. This clamshell roof allows the team to adjust the cars by simply lifting the lid. The experimental sensors carried by the cars, and the cars themselves, are first-of-their-kind prototypes, and thus require occasional adjustment for optimal performance. Therefore, ready access to the cars was an invaluable feature that eventually became a design requirement. This garage assembly is at the end of the insertion arm, next to the hump, which is the part that hooks to the end of the internal cask lid to align the robot with the vent it inspects. The hump is a rounded component with a special geometry to prevent wedging and jamming of the car system during cask inspection. The final garage, hump, and arm prototype is shown in Figure 6.
Figure 6: The insertion arm segment of the robotic inspection system, specifically the garage segment. The image on the left shows the garage closed, and the image on the right shows the garage open to display the cars housing the sensors. The cream-colored component at the center top of the left image is the hump. The center-bottom aluminum component in the left image is the garage lid.

4.2 Key Examples of Changes that Spurred Many Knock-On Changes

The design of the garage provided many examples of incremental changes, and in particular showed changes that appeared incremental but spurred a multitude of “knock on” changes. The first design of the arm system was quite simple. It involved steel sheet metal as a ramp, and a curved hump on the end that was approximately the size of half of a coffee can. This evolved into an aluminum ramp and a hump made via additive manufacturing. This early design was very simple to fabricate and use, but it allowed the cars to slide off the arm during
insertion. The car-containment problem was solved by the addition of a garage, but the stationary design of the first garage did not allow operators to access the car without having two assistants. This did not meet the project goals, and so the garage was redesigned again to be a clamshell style. This design of the garage met the project needs well, but required redesign of many other components to allow the insertion arm to function in other ways. This change to a clamshell design is the second of the examples discussed later that exemplify successful implementation of the new method of managing change propagation.

One of the incremental changes that caused a need for many “knock on” changes was a redesign of the garage to allow the robot car train to be extracted quickly from the garage. This design change was the addition of a spring reel, and it will be the subject of the first example of the proposed change propagation method. This spring reel was added to the insertion arm to extract the cars from the garage and force them up and over the hump. It was necessary because the original plan for extracting the cars involved pushing them by applying force to the tether from outside the cask. This method did not result in easily repeatable car insertion. This was unacceptable because measurement accuracy of the robot cars’ position is negatively affected by any inconsistency in insertion.
The addition of a spring-reel, pictured in Figure 7, illustrates how the traditional component connection DSM missed interdependencies; the DSM for the system prior to adding the spring reel suggested that only certain parts of the system would be affected by this change. Specifically, it can be seen in Figure 8 that Components 14 and 15 are not attached in any way to Component 11 or 12, which is where the new spring reel was added. Therefore, this addition was not expected to create issues with connections associated with Components 14 and 15.
Figure 8: DSM of the garage portion of the insertion arm before the addition of the spring reel.

Figure 9: DSM of the garage portion of the insertion arm after the addition of the spring reel. Green boxes with represent connections added upon implementation of the spring reel, orange boxes with represents connections where problems occurred due to the addition of the spring reel.
Analysis of the garage-component DSM after the design change shows how the pre-change DSM appears to lack pertinent information. After the addition of the spring reel, the post-change DSM revealed that changes had to be made for Components 14 and 15, which was unexpected. In fact, the spring reel and cable were intentionally physically placed in the updated design to avoid interactions with the back of the garage where Components 14 and 15 are located. However, the spring reel did affect those parts, and it caused significant redesign efforts. This raised the question of how this change could have been predicted; this unexpected result was one of the events that spurred exploration of a different DSM method to manage changes during prototyping.

The DSM in Figure 10 shows these affected connections boxed, to illustrate how at the first level DSM, it would not necessarily be obvious that these components would change. These components do have secondary connections, because they connect to components that in turn connect to the ones being changed, but there are many other components with secondary connections that did not change. Thus, without a far more complex DSM method perhaps involving likelihood of change and secondary connections [23], the resulting change in Components 14 and 15 would not have been seen as probable. This led to interest in finding a method that would show which components might change when implementing a top-level connection DSM change.
This unanticipated cascading of design changes, i.e. “knock on” changes [10], led to the aforementioned conclusion that either a more elaborate DSM would need to be constructed, one that included different connection types, or a different type of change propagation method would be useful for this project. The literature was explored to find change propagation methods that would suggest changes like the ones mentioned in Figure 10 in a simple-to-implement form, but the closest items we found were DSM matrices with multiple connection styles and DSMs with delta DSMs to show changes [9]. These were helpful for tracking changes, but involved a more complicated method to implement within a DSM than was desired for the sake of rapid implementation. Existing methods also often required experienced designers to give feedback on probability of changes, and the difficulty of changing components, based on their previous experience [7]. The PRINSE NEUP project involved many design uncertainties because of the
number of systems that had never before been implemented; therefore, a method was needed to manage connections with only the information at hand and one that could be rapidly implemented.

4.3 Overview of the New C+C DSM Method

Using the prototyping efforts of the PRINSE project as a case study, the missing factor in the DSMs was determined to be that it did not capture the constraints on the components other than their connectivity. The connections that were being broken that were unanticipated were all found to be linked to overarching system design constraints. This insight led to a new DSM method developed in this thesis to specifically manage and illustrate both physical connections and constraints easily. This method, hereafter named the C+C DSM method, with the abbreviation C+C standing for Components and Constraints, has the following steps:

New C+C DSM Method Steps:

1. Create a list of system components
2. Create a list of design constraints for the system
3. Combine these into one list
4. Use this list as the rows and columns for a DSM, and mark physical connections
5. Mark connections between constraints and any components that they affect in the same manner as physical connections.
6. Sort the matrix to define modules and divide work among parties involved
7. Add or remove components/constraints as necessary
8. Resort (cluster) matrix as necessary
These steps are similar to those used to form a conventional DSM method, except for Step 5. In this step, the C+C DSM method effectively treats all parts connected to shared constraints as primary connections. This direct interconnection of parts through constraints is different from the interconnection of two parts through an intermediate part (i.e., a second level connection between a primary part and another part) because when a constraint is violated, there are no intermediate parts to absorb the constraint. In contrast, a second-level connection allows an intermediate part to absorb physical changes and thus may block or absorb “knock-on” effects.

4.4 First Example of the C+C DSM Method

Using the new C+C DSM method, the same example of changing to a spring reel can be examined again to see if it would have predicted changes to Components 14 and 15. This matrix includes components, constraints, and connections between constraints and components as per the new method. It can be seen in Figure 11 that Components 14 and 15 might be affected, because the components being added are affected by constraints to which those two components are connected. Figure 11 shows the C+C DSM matrix before the design change was made, with the newly revealed connection of Components 14 and 15 to constraints boxed. For ease of comparison, the original DSM without constraints is shown on the same page in Figure 12, also with Components 14 and 15 boxed, so that the connections in the physical connection DSM can be compared to those in the C+C DSM Method.
Figure 11: C+C DSM of the garage portion of the insertion arm before the addition of the spring reel. The constraints are now included in the matrix in a manner identical to components, as per the new method for tracking design change propagation. The interactions between Components 14 and 15 and the rest of the system are boxed.

Figure 12: Conventional DSM of the garage portion of the insertion arm before the addition of the spring reel, without constraints. The connections for Components 14 and 15 are boxed, and can be seen to not include components 11 or 12.
Figure 13: C+C DSM of the garage portion of the insertion arm after the addition of the spring reel. Green boxes, denoted “o” for origin of a new connection, represents connections added upon implementation of the spring reel, orange, denoted “p” for possible problem, represents connections where changes occurred due to the addition of the spring reel. Connections of Components 14 and 15 are boxed. The constraints are all included as per the new C+C method.

4.5 Second Example of the C+C DSM Method

To reinforce the efficacy of the C+C method, another key example is presented from the PRINSE project. At a later stage of the garage design, the lid of the garage was interfering with the user’s ability to access the robot. In fact, it was impossible to remove the robot from the garage with fewer than three people, or without significant disassembly of the garage. This was an undesirable situation because regular access to the robot was needed to adjust sensors and to make periodic changes to the cars.
To fix this problem, the design team chose to modify the garage lid so that it could be opened via a clamshell-style hinge, allowing the team rapid and simple access to the cars. This seemed like a small change during the ideation and analysis of this modification. However, the implementation spurred many other “knock-on” design changes.

To illustrate how such changes would have been difficult to predict using conventional DSM approaches, the component DSM is presented. It is then compared to the C+C DSM method, and both are shown in Figures 14 and 15. Comparing the physical connection DSM with the C+C DSM, it is apparent that components in the C+C DSM are far more interrelated than would be expected from the conventional physical connection DSM. The constraints show the density of connections between Components 10 through 25 is far higher than the conventional DSM suggested. For example, Constraint 4, the accessibility of the robot, is related to every component from 10 through 18.
Figure 14: Conventional DSM of the system before changing to the clamshell garage lid.

Figure 15: C+C DSM from before the clamshell garage lid change occurred, including constraints in the C+C style.
To visually illustrate how the propagated changes in the system would be difficult or impossible to predict if constraints were not included, the physical connection DSMs for both before and after the clamshell lid modification occurred are pictured in Figure 16 and Figure 17. For the clamshell redesign, the only component that was intentionally changed was the lid, which is enumerated as Component 11. In the “after” DSM, the green and orange connections all represent changes, and it is immediately apparent that many of them are not connected to Component 11 at all in the “before” situation, which was the only component changed intentionally. However, the DSMs from before and after the clamshell modification show that component connections were affected for more than half of the system components. Out of components that were changed, Components 5, 13, 14, 17, and 23 were not connected to the component being directly modified, namely Component 11. These parts are thus experiencing unexpected changes as they were not predicted using only the conventional physical-connection DSM.
Figure 16: C+C DSM with only physical connections, prior to the garage lid modification to a clamshell design. Connections of the component that was purposefully changed to create the clamshell design are boxed.

Figure 17: DSM of the insertion arm system after the clamshell garage lid modification. Connections that became problematic are orange with the letter “p” for possible problem, those that were added are green with the letter “o” for origin of connection. The connections that were modified on purpose are boxed.
Figure 18: C+C DSM created to show connections in the system using the C+C method, before the change to the clamshell garage lid design. Red boxes are around the constraint connections affected by the change to a clamshell style garage lid. Component 11 was the only component intentionally modified, and it is boxed in orange.

It is apparent that many of the changes in the system shown in Figure 17 would not be expected from the DSM shown in Figure 16. This leads to the question of whether the C+C Method would improve the ability to predict change propagation. Examining the C+C DSM in Figure 18, one can observe that all of the affected components are related to the constraints affected by the change. Specifically, Figure 18 shows that more changes were likely than the physical connection DSM in Figure 16 suggested. The boxes show both the constraints and the component that was changed, which clearly include far more connections than the physical connection component DSM alone would have included. Now that constraints are included, as
per the C+C DSM method, any component and constraint connection that is changed reveals the whole component and its connections may also be subject to changes.

Once the constraints have been added, the components at risk can be seen to be any that are connected to the affected constraints and thus possibly affected by the change (ergo those labeled orange or green), which happen to include every component that experienced connection changes. This effect can be seen in Figure 19. Specifically, this clamshell modification to the garage lid design caused 13 components to have connections change or need to be changed. This is not predicted by examining the physical connections to Component 11 in the DSM in Figure 16 right before the change was made. However, using the C+C DSM Method, it is apparent that 14 components had potential to change, because they were connected to the constraints affected by the change. This reveals that the C+C DSM in Figure 19 gives a more accurate prediction of components that might be changed, because every component that ended up having a connection change was identified by the C+C DSM Method. This is a sharp contrast to the DSM in Figure 16 representing the system before the change, where if the designer looks at only what was purposely modified to include the clamshell garage lid redesign, she/he would expect only Component 11 and those connected to it to be affected by the redesign. That would not be accurate. Figure 19 shows that the C+C DSM Method correctly predicted all thirteen components that ended up changing, as well as indicated one additional component that might have changed, but did not.

In analyzing the C+C DSM Method, it is notable that this method allows design constraints to replace secondary, tertiary, etc. physical connections with primary-level connections. As reviewed in the literature in Chapter 2, there are many methods for examining secondary connections, and even tertiary connections. Secondary and tertiary connections could
have shown changes needed in the components that were affected, but the specific primary and secondary connections affected would have been difficult to predict from simply looking at the first matrix in Figure 16. The C+C DSM method removes the need to examine these secondary and tertiary connections.

The example shown in Figure 19 highlights an important nuance of the C+C DSM Method, which is: when constraints are activated as part of a design change, all components interacting with that constraint are immediately at risk of requiring changes as well. Thus, a constraint between two components could make these two components – which may not be physically connected at even secondary or tertiary levels – interact within a DSM structure in the same way as components directly connected to the changed components at the primary level.
This C+C DSM Method readily identified dependencies that were hidden in the purely component-based DSM. It can be seen in the first example of the spring reel addition that Components 14 and 15 are connected to Constraint 2, which would be connected to the spring reel when it was added, thus predicting that Components 14 and 15 might be affected by this design change – which they were. This meets the first goal defined in Chapter 2 for this method: to predict which parts have potential to be affected by changes via looking at DSM connections. This can be seen by how all of the “knock-on” changes can be predicted using this C+C DSM method in this simple connection matrix in Figure 11 and Figure 15. These figures also show how the new C+C DSM Method meets the fifth goal outlined in Chapter 2 for the new method:
to account for not only components, but for their relations to constraints imposed by the
implementation environment. This is achieved by including constraints as items in the DSM
rows and columns. These dependencies add value because they predict “knock on” changes
more accurately than the original DSM. The C+C DSM Method shows connections between
parts and constraints without adding more connection types or levels that would increase method
complexity.

It is also apparent that the C+C DSM Method meets the second goal of showing which
parts will be more time-consuming to redesign, based upon the metric of the number of
constraints and components to which each part is connected. The C+C DSM Method also
provides insights similar to conventional DSM methods; for example, the more component and
constraint connections any part has, the more time and care must be taken with its redesign
process. With this logic, the addition of a spring reel would be predicted to be complicated,
because though it was added directly to two components with few connections, it was also
affected by all four constraints, which have a plethora (more than 35) of connections.

Additionally, the C+C DSM Method stays true to the classic feature of DSMs in that it
suggests how one might divide work among parties involved, if sorted in a manner typical to
DSMs, such as clustering connections around the diagonal. This meets the third goal of the new
C+C DSM Method: forming sub-groups or modules, and the fourth goal of showing where they
need to communicate.

One example of how the project could be sorted using this methodology can be seen in
Figure 20. This matrix represents the system before the addition of the spring reel. This sorting
was done using in MATLAB using the algorithm developed by Thebeau [23]. Two example
outcomes of this sorting can be found in Appendix A. Interestingly, the sorting algorithm
produced subsystem clusters that show this portion of the project was divided almost appropriately; the two groupings suggested when including constraints match those used in the project - except for the placement of two components, the hump top and hump bottom. It is important to note that this algorithm can converge to multiple different solutions, each representing an equilibrium without knowing which one is the global optimum. Partly due to this situation, engineering judgement is the final deciding factor in grouping selection for modules or team tasks. However, the agreement between the optimization and the inherent team structure does suggest the C+C DSM Method could provide insight valuable in guiding a team’s activity and organization.

Figure 20: DSM created for the insertion arm before the addition of the spring reel. This DSM has been sorted using a MATLAB algorithm to form two modules, which are shown with the red boxes.
This example, while only one instance of tracking changes during prototyping, shows how that the proposed C+C DSM Method for managing change propagation met the requirements for a change propagation method that were outlined in Chapter 2. This work also shows that the new method for managing change during prototyping was effective in predicting changes for parts of the PRINSE NEUP project, and these positive results suggest other similar projects might benefit from this knowledge.
Chapter 5

Thesis Outcomes and Future Work

There are several takeaways from this new C+C DSM Method for managing change propagation during prototyping. The first takeaway is that the C+C DSM Method can predict which components may be affected by changes with only one type of connection. The second takeaway is that only one level of connections is required with the proposed C+C DSM Method, hence, no examination of secondary connections is needed to predict components that may change. The third takeaway is that the C+C DSM Method can assist in determining which parts of the system are likely to require a lot of time to redesign. It can also, in the style of DSMs, show how the project can be divided and where those sub-teams involved across various segments will need to communicate. Finally, this work has shown that implementation environment constraints and their relationships to components can also be represented with this method, in the same manner as components. These takeaways match those desired originally in Chapter 2 of this thesis, when the goals for the new method were outlined.

Building off this work in the future, the proposed change propagation method for prototyping would benefit from further experimentation. This method was created and tested via application to prototyping history after the completion of a project, based upon experiences during that project. It was not employed during the main prototyping phase of the project to actively manage changes and make decisions, though it was used at the end of the project during post-mortem analysis. Future work could therefore include case studies of the proposed method application in practice from start to end of projects, and it would be particularly useful to
perform side-by-side comparisons of this method in practice on the same design iterations, to compare C+C DSM to conventional DSMs for their design insight.

Additionally, while the previous chapter showed the efficacy of the C+C DSM Method for managing change propagation on a system design physical prototyping project, it was on a subsystem of that project. This situation meant some constraints could be component connections that would be shown in the full system DSM, such as the vent height, but that are omitted for the subsystem analysis. Some constraints related to functionality, such as the ability to remove the cars from the garage, would remain as constraints even in the total system DSM. In future work, these situations in which something could be either a constraint or a component connection to an external system could be tracked for comparison purposes. The use of constraints to analyze a subsystem rather than the entire design of a complex system could significantly reduce computational cost due to decreasing the size of the matrices.

Another potential case study would be application of this method in an academic environment to see if it improves outcomes for novice designers during the construction of their prototypes. Novice designers’ lack of experience makes them susceptible to missing connections between components, so the proposed method might assist them in managing change propagation. The benefit of the C+C DSM Method could be tested by comparing the outcome of prototypes made without a change management method to the outcome of those made with the proposed C+C DSM Method, in a manner similar to that of the tests done with engineering students when examining the efficacy of Prototype for X (PFX) [24]. This method might benefit novice designers as it makes use of knowledge they can easily obtain about how their prototypes work, such as whether or not parts are physically connected, and it clearly illustrates their overarching project design constraints.
Testing this C+C DSM Method with industry prototyping projects would also be a valuable direction for future work, as the effect of the method on experienced designers may vary from that of novice designers. It is apparent that the new C+C DSM Method does predict changes that the DSM based purely on physical/spatial connections between components did not predict, which might decrease the effect of experience during the creation of prototype systems. A possible study of this method could involve examining the gap between experienced and inexperienced design teams’ prototype outcomes, to see if use of the proposed method in this thesis decreases the discrepancy between those two groups.
Bibliography


