Application and Management of Commonality within NASA Systems

by

Richard Alexander Rhodes

B.S. Mechanical Engineering, The University of Texas at Austin, 2008

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

MASTERS OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

[February 2010]
January 2010

Signature of Author: ________________
Department of Aeronautics and Astronautics
January 29, 2010

Certified by: ____________________
Edward F. Crawley
Ford Professor of Engineering
Thesis Supervisor

Accepted by: __________________________
Prof. Eytan H. Modiano
Associate Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Application and Management of Commonality within NASA Systems

by
Richard A. Rhodes

Submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of Masters of Science

Abstract

Commonality can be defined as the sharing of assets such as components, designs, processes, technologies, interfaces, and/or infrastructure across systems. Through commonality, NASA has the opportunity to develop, produce, and operate systems more efficiently, thus reducing their life cycle costs. However, the benefits gained from commonality greatly depend on how commonality is managed, i.e. how successfully the identification of opportunities is managed and how those opportunities are subsequently evaluated and implemented.

The goal of this research is to observe how commonality is managed within NASA systems and to identify the best practices and key challenges for the management of commonality. To that end, three case studies were conducted of past and present NASA systems: the Constellation Space Suit System (CSSS), the International Standard Payload Rack (ISPR), and the Core Flight-Software System (CFS). Each of the case studies was chosen because it offered a diverse view into the management of commonality, by differing in the program in which the system was developed, the type of system that was developed, and the method used to develop the system.

The case studies offer a detailed look into current management practices within NASA and allow for comparisons to be made with seven industry case studies, previously conducted by Boas (2008). The three NASA case studies showed that two trends that were consistently observed in the industry cases (life cycle offsets and divergence) also exist within NASA systems. In addition, the management approaches were observed to be common between NASA and industry. In conducting the NASA case studies, several management methods were identified that can encourage the successful application of commonality, including: the organizational structure, the level of management support, the type of contract, and design competencies. Each of the observed management methods are discussed in the thesis within the scope of the individual case study and in the broader scope of all three systems in the cross-case analysis.

In each of the three NASA case studies and seven industry case studies, it was observed that the evaluation of commonality was often either overlooked or reduced to notional or qualitative analysis. To address this problem, both a deterministic economic cost model and an economic cost model that evaluates commonality as a Real Option were developed to better evaluate the systems. Evaluating commonality as a Real Option allows management to consider uncertainties and flexibility in the system. Both methods of evaluating opportunities for commonality are applied using information from the CSSS case study as an example.

Thesis Supervisor: Edward F. Crawley
Title: Ford Professor of Engineering

Acknowledgements

First, I would like to thank my family. The continuous support and encouragement of my parents, Richard and Kate, and sisters, Karyn, Kristyn, and Carey, was invaluable throughout my life and in particular my graduate career. Britney, my fiancée, has made more sacrifices to allow me to pursue a graduate degree than I could imagine. She has been an irreplaceable friend and supporter in my life for the last four years and I am eager to spend the rest of my life with her.

I would also like to thank Edward F. Crawley, Ford Professor of Engineering, for providing me with guidance and support through out my graduate studies. Ed is a brilliant engineer and a gracious advisor. Ed offers a unique perspective, technical savvy, and decades of experience that has guided my studies the last few years.

This research would not have been possible without the guidance and assistance of Prof. Jeff Hoffman, Prof. Oli de Weck, Prof. Richard de Neufville, Nantel Suzuki (NASA), Raul Blanco (NASA), Nicole Jordan (NASA), Johathan Wilmot (NASA), Raphael Grau (NASA), Gary Spexarth (NASA), and all of the NASA engineers and managers that took the time to work with me and discuss the application and management of commonality within their systems. In addition, Kathi Brazil has been a good friend and always worked hard to ensure I was able to get time with Ed.

Finally, I would like to thank my friends and fellow graduate students for their encouragement and insight, including: Arthur Guest, Wilfried Hofstetter, Chase Cooper, Anthony Wicht, Matt Silver, Alessandro Golkar, Bruce Cameron, Daniel Selva, Brandon Suarez, Howard Yue, Maokai Lin, Wen Feng, and Ryan Boas.

Thank you!

Table of Contents

Table of Figures

Table of Tables

1. Introduction

The creation of a large engineering system requires decision makers to prioritize between competing goals. These goals include, but are not necessarily limited to, cost, technical performance, schedule, and risk. The listed goals are competing because they cannot all be optimized at the same time. The most common representations of this trade are the Triple Constraint Concept (NASA Cost Estimating Handbook, 2008), and the Iron Triangle of Management. Applying commonality to a system offers one method of trading these goals in order to offer a project or system-wide benefit; often involving a trade in performance to offer either schedule, risk, or cost benefits.

Commonality can be defined as the sharing of assets such as components, designs, processes, technologies, interfaces, and/or infrastructure across systems (Boas, 2008). Commonality is not the only method of improving a system, but through commonality NASA has the opportunity to develop, produce, and operate systems more efficiently. The specific benefits from the application of commonality as they apply to each of the life cycle phases are discussed in section 2.4 Benefits and Penalties.

Commonality is not a new concept, but has been implemented by industry and the government for many years as seen in power tools, printers, cars, airplanes, and satellites (Boas, 2008). For example, the Department of Defense (DoD) recently implemented commonality in the design of a new fighter jet, the Joint Strike Fighter (JSF) or F-35. The JSF was developed for use by the Air Force, Navy, and Marines, each of which have different requirements for their fighters, as seen in the variants in Figure 1. Application of commonality in the JSF project allowed the government to procure three aircraft variants for the price of 1.8 aircrafts (Boas, 2008).

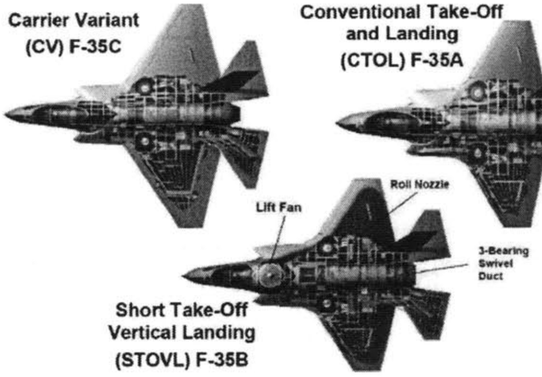


Figure 1: Joint Strike Fighter design variances, (from (Boas, 2008))

Commonality is created between systems by the reuse of assets from a previous system. The reuse of assets can be conducted in two ways: the asset can be reused reactively, without prior planning; or the asset can be developed in the first system with the intent that it becomes common to future systems, i.e. forward commonality. In order for commonality to have substantial, widespread benefits to a system, forward commonality should be actively sought out and applied. However, the benefits gained from commonality greatly depend on how commonality is managed.

The goal of this research is to observe how commonality is managed and applied within NASA systems and to identify the best practices and key challenges associated with the management of commonality. To that end, three case studies were conducted of past and present NASA systems: the Constellation Space Suit System (CSSS), the International Standard Payload Rack (ISPR), and the Core Flight-Software System (CFS). Each of the case studies was chosen because they offer a diverse view into the management of commonality by differing in the program in which the system is operated, the type of system that was developed, the organizational structure, and the method used to develop the system. However, the theoretical most difficult management case was not observed in the NASA case studies, in which commonality must be managed between two contractors. The remainder of this thesis is organized as follows:

Chapter 2 presents much of the current knowledge on commonality, including a discussion of the challenges for application, management approaches used, and the benefits and penalties of commonality to a system.

Chapter 3 then presents the first of the three case studies, on the Constellation Space Suit System (CSSS). This chapter includes a background to the Constellation Program, a discussion of the architectural requirements, the development process used, and finally observations on the system challenges and the management of commonality.

Chapter 4 follows a similar format for the second of the three case studies, on the payload management system with a focus on the International Standard Payload Rack (ISPR). The chapter contains a discussion of the background of the system, the original architecture of the system, the evolution of the architecture, and observations on the challenges and management methods used in this case.

Chapter 5 then presents the findings from the third case study, on the Core Flight-Software System (CFS). The chapter contains a background to the system, information on the architecture of the system, the evolution of the system, and finally observations on how commonality was managed within the system.

Chapter 6 presents the cross-case analysis and final conclusions from all of the case studies. This chapter also presents guidance for NASA managers on how to best manage commonality; including different approaches for applying commonality and specific management methods that should be used based on the type of system, the development method, and the organizational structure of the particular system.

Commonality does not always benefit a system, but often implies a trade between competing goals. Therefore, opportunities for commonality should be evaluated to determine the benefits and penalties. In the NASA and industry case studies, the evaluation process was predominately either not conducted or limited to a qualitative analysis. Chapter 7 presents the method and application of two quantitative methods of estimating life cycle costs of common systems: a deterministic economic cost model and an economic cost model that evaluates commonality as a Real Option. Evaluating commonality as a real option allows managers to consider uncertainty in the system and the value of flexibility in order to develop the optimal development strategy.

Finally, Chapter 8 summarizes the contributions of the thesis in order to create a better understanding of the management of commonality. Recommendations are made for future research.

2. Introductory Concepts and Background Literature

This chapter presents the introductory concepts and background literature that serves as the basis for the NASA case studies. The three case studies of NASA systems were intended to build on the seven industry case studies previously conducted by Boas (2008), as a result this chapter discusses many of the findings from the seven industry case studies, as well as findings from other literature on commonality management.

The first section, Types of Commonality, more accurately defines the concepts of commonality that will be researched in this thesis. Next, two trends that were identified in the industry case studies, divergence and life cycle offsets, are discussed. The discussion includes the effects that these trends have on the management of commonality in practice. Next, the critical management tasks are explained along with observations from the industry case studies on the different approaches to manage these tasks. Finally, the benefits and penalties of commonality are discussed as they relate to each phase of a system's life cycle.

2.1 Types of Commonality

Commonality can be defined as the sharing of assets such as components, designs, processes, technologies, interfaces, and/or infrastructure across systems (Boas, 2008). While the definition of commonality remains constant there are two distinctly different methods of applying commonality between systems: reactive reuse and forward commonality.

Reactive Reuse

Reactive reuse is the case in which a system reuses a component that was designed solely for a past system by integrating the component into a new system. Reactive reuse is not associated with any forward planning or coordination, and as a result can be limited in scope compared to forward commonality. However, reactive reuse is a common practice with significant benefits.

Forward Commonality

Forward commonality is the case in which opportunities for commonality are actively sought out and developed in the first system with the intention that they become common and are reused in the future system. Forward commonality is referred to under other names in other studies, such as the development of product families and design for reuse. This study will primarily focus on observations and good practices for the management of forward commonality because it allows for the maximum benefits from commonality to be obtained.

There are additional terms that are worth defining at this time because they are related to aspects of forward commonality and were observed in the NASA case studies: modularity and standardization. *Modularity* is defined by Fricke and Schulze (2005) to be a principle that aims at "building a system architecture that clusters the system's functions into various modules while minimizing the coupling among the modules (loose coupling) and maximizing the cohesion within the modules (strong coupling)." Modularity is developed for several reasons, but one of the benefits is that it makes a system flexible by allowing modules to be swapped out or allowing the same module to be shared by multiple systems, as seen in Figure 2. Modularity and commonality are closely related because all types of modularity require at minimum a common interface and often result in additional commonality. Component sharing modularity was utilized in the Constellation Space Suit System (CSSS) case study in order to obtain life cycle cost benefits and to reduce the launch mass of the system. Component-swapping modularity was utilized in the International Standard Payload Rack (ISPR) case study to offer additional flexibility.

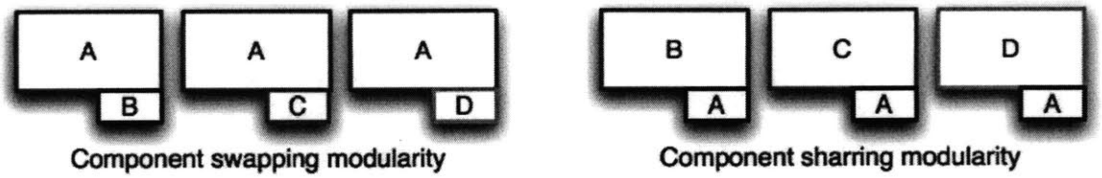


Figure 2: Two of the six types of modularity (altered from (Fricke & Schulz, 2005))

Standardization is the creation of some degree of similarity or commonality by an authority or general consent. Standardization in all cases creates some degree of commonality and is referred to in some cases, including NASA documents, synonymously with commonality. However, standardization is often implemented at lower levels of the system, such as the component or interface level, and predominately benefits production and operations. Standardization can be a useful method of applying commonality between peer organizations in which no central authority exists to coordinate or manage designs. Standardization was used to apply commonality across systems in both the CSSS and ISPR case studies.

2.2 Trends in Commonality

Divergence and life cycle offsets were observed in each of the seven industry case studies and have been shown to dramatically impact the management of systems. Therefore, the first goal of the NASA case studies is to determine whether or not these trends also exist in NASA systems.

Divergence

Industry case studies conducted by Boas (2008) showed that commonality consistently decreased or diverged from what was originally planned. The trend that was commonly seen in the industry case studies was an increase in the amount of intended commonality during the planning phase, but a consistent decrease in commonality as development progressed, depicted by Figure 3.

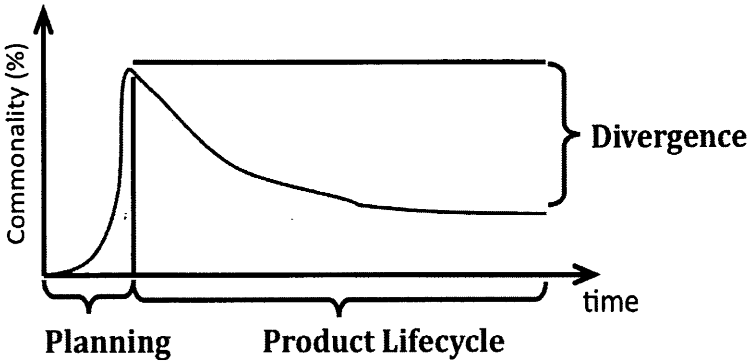


Figure 3: Divergence in a system's life cycle

Commonality should not be the final goal of a system because commonality does not always benefit a system. Instead, the goal should be to obtain the maximum benefit from commonality. Therefore, divergence, or a decrease in commonality, can occur to either benefit or penalize a system. Acceptable factors cause divergence in order to improve the system, while unacceptable factors cause divergence while penalizing the system. The causes for divergence as observed in the seven industry case studies (Boas, 2008) include:

- Acceptable factors:
  - Market change – Demand for the common item drops, requiring developers to seek unique solutions; change in requirements
  - Technology – New technology has become available that can improve the product if divergence occurs; technology has become obsolete
  - Learning – Learning in development may show that a unique solution is better than the already developed common solution because of technical factors or economic factors
- Unacceptable factors:
  - Poor management or lack of coordination
  - Intentional pursuit of uniqueness
  - Failure to consider life cycle benefits

Life Cycle Offsets

Much of the literature and scholarship on commonality assumes that the development of common systems is conducted in a parallel and coordinated manner. However, seven industry case studies, previously conducted by Boas (2008), found that in most cases life cycle offsets exist. Meaning that while planning for the two systems may be conducted at the same time, the development, production, and operations of the common systems are offset in time, as seen in Figure 4.

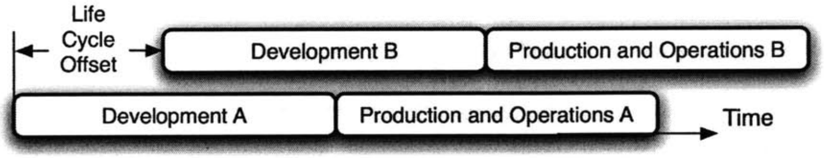


Figure 4: Life cycle offsets often occur between two common systems

A few factors were identified by Boas (2008) to have the tendency to create life cycle offsets:

- Market factors - testing market with first variant; or different dates of need
- Technology factors - technology capability development; and learning from earlier products
- Organizational factors - organizational focus on a product; and human resource constraints
- Financial factors - total program cost; cash flow management; and budgetary restrictions

Commonality in Practice

Life cycle offsets and divergence have several impacts on the management of commonality. This section discusses the effects that both divergence and life cycle offsets have on the management of common systems.

Offsets create a necessary decision at the time of development: to either develop with intended commonality or to develop independently

Life cycle offsets between systems result in decisions that must be made at the time that the first system is developed: components can be developed with the intention that they become unique, or with the intention that they become common, as indicated at Time 1 in Figure 5. However, divergence and reuse often occur. Therefore, at the time that the future system is developed, common components either become common as planned or unique to the first system because of divergence, indicated by classes 2 and 4 respectively in Figure 5. Components that were developed with the intent to be unique to the first system are either reactively reused and become common or remain unique to the first system, as indicated by classes 1 and 3 respectively in Figure 5. Based on this framework developed by Boas (2008), there are five classifications of components between common systems. The five commonality classes form the basis for the economic models discussed in Chapter 7.

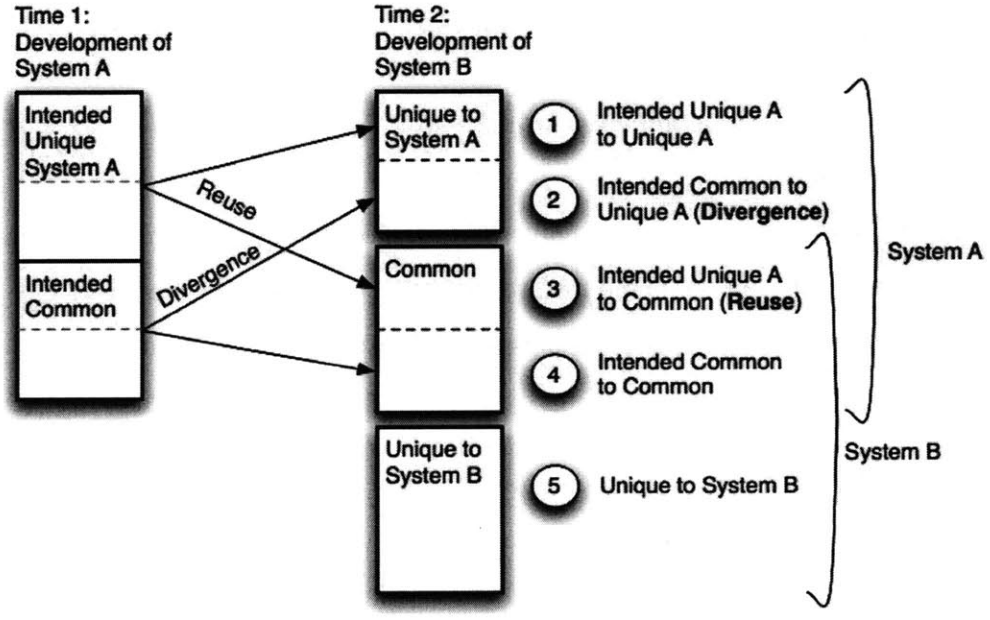


Figure 5: Framework for classifying commonality, which takes into account life cycle offsets, divergence, and reactive reuse (based on (Boas, 2008))

Offsets cause an up-front development penalty

Developing components with the intent that they become common with future systems involves developing the components to meet additional requirements; the additional requirements result in additional development costs. In the benefits and penalties sections the up-front penalty is described as the *cost of common development.*

Existence of offsets cause benefits to be offset to future systems

While the first system inherits an up-front development penalty, most of the development benefits are passed on to later variants in the form of reduced development work. The later variants do, however, inherit a development cost associated with the integration of the common components into the system, i.e. the *integration penalty.*

Total benefits from commonality decrease as a result of offsets

Offsets in production and operations decrease the potential benefits obtainable from commonality, including *economies of scale* (production), *capability overlap* (operations), and *learning* (production and operations). Each of these factors are discussed in more detail in Section 2.4 and Chapter 7.

Offsets cause future variants to be uncertain and often unrepresented at the time that the first system is developed

At the time that the first system is being developed, the later variants are uncertain. As a result, design decisions are weighted towards the current system. This issue is compounded by the fact that future variants are also often unrepresented at the time that decisions are made.

Offsets increase the likelihood that divergence will occur

Life cycle offsets increase the likelihood that divergence will occur because there is a greater likelihood that the market or technologies will change as more time passes.

2.3 Management

In the industry case studies conducted by Boas (2008), the management approach and methods used for the application of commonality was found to be a critical factor for success. This section begins by discussing the critical tasks associated with the management of commonality, which include: managing the identification of opportunities for commonality, evaluating opportunities, and implementing beneficial opportunities for commonality. Next, the management approaches observed in the industry case studies are discussed.

Management Tasks

In conducting the NASA case studies, it was observed that in order for commonality to be successfully applied to a system, the opportunity for commonality must be: (1) technically feasible, (2) beneficial, and (3) implemented successfully. Therefore, in order for commonality to be managed successfully there are also three critical management tasks:

1. The management of the *identification* of opportunities
2. *Evaluation* of opportunities to determine whether or not each opportunity is beneficial
3. *Implementation* of the opportunities that are deemed beneficial

Each of these management tasks presents challenges that must be overcome. The first task is to manage the *identification* of opportunities. For the first task, the challenge lies in the fact that forward commonality will not be created by chance, but must be actively sought out and identified. This study will not focus on the technical feasibility of commonality, but instead on how the identification of opportunities is managed.

The second task is to *evaluate* opportunities for commonality to determine the benefits and penalties. Commonality will not benefit the system if it is blindly implemented wherever technically feasible. Instead, each opportunity should be evaluated to determine the benefits and penalties that it will have on the system. Commonality has many benefits and penalties that affect the system in different and sometimes unexpected ways, making the evaluation of opportunities difficult. The benefits and penalties are examined in detail in Section 2.4, Benefits and Penalties. In addition, the work conducted to improve the evaluation of opportunities for commonality is discussed in Chapter 7.

The last task is to *implement* opportunities for commonality that are deemed beneficial. Opportunities for commonality are often not implemented as planned, because there are several challenges, including life cycle offsets and divergence.

The management methods used in each of the three NASA case studies is discussed at the end of each chapter in relation to each of the management tasks identified in this section. The final cross-case analysis discussing each of the management methods is discussed in Chapter 6.

Management Approaches

In Boas' study of industry applications of commonality (Boas, 2008), three levels of commonality management were observed: (1) informal reactive reuse, (2) the development of common building blocks, and (3) a formalized process for the widespread application of commonality. Each of the observed levels are described and expanded on below. The management approaches observed in the three NASA case studies is discussed in Chapter 6.

Reactive Reuse Approach

There are a few aspects of each of these levels that distinguish them from each other. In the case studies conducted by Boas the most common level of commonality management observed was the informal reactive reuse of components. This approach is described by a single-product culture in which the focus of development is on the optimization of a single product at a time. Instead of developing forward commonality, these organizations often develop independently and reactively reuse assets from previous systems as they see fit. The most beneficial form of reactive reuse involved the use of the previous system as the baseline for future development. In addition, little to no formal ownership of commonality and no processes to control commonality and changes to the system were observed.

Common Building Block Approach

The other two levels of commonality management were observed less often than the first and both of these levels applied aspects of forward commonality. The second level of commonality management still predominately involves single-product cultures, however items of value were developed separately as common building blocks to be used on future variants. The common building blocks were often managed outside of the control of any individual project in a parallel development stream.

Widespread Forward Commonality Approach

The third level of commonality management that was observed is described as the creation of a formal process to identify, evaluate, and implement widespread commonality. At this level, widespread forward commonality is actively sought out. Common components are specifically identified as common and there is ownership for the common components. There are also formal methods in place to control changes to the system. Boas describes the ultimate approach for commonality management as the development of a commonality culture. The commonality culture is the approach in which commonality has been so engrained in the culture that a formalized process become less important. The commonality culture is viewed as the ideal case and an extension of the widespread application of forward commonality by extensive management support.

In each of the levels of commonality management described, the development of control processes and ownership of commonality were consistently associated with the successful management and application of commonality. Each of the observed methods and aspects of management are highlighted in Table 1.

Table 1: Summarization of the various levels of commonality management, as defined by Boas (2008)

	Informal Reactive Reuse	Common Building Block	Widespread Formal Process
Culture	Single product culture	Single product culture w/ a few exceptions	Common product culture
Development Process	Previous product is starting point for future variants	Independent development with effort placed on commonality for a few high-value components	Formal process to identify, evaluate, and implement forward commonality
Ownership of Commonality	No formal ownership of commonality	Formal ownership and tracking of a few common building blocks	Formal ownership and tracking of commonality
Other Comments	Forward commonality may have been considered in some contexts, but by far the dominant effort was on a single product	Common building blocks were developed independent of a particular program in a few cases	At times commonality was implemented purely for commonality sake, without proper evaluation techniques

2.4 Benefits and Penalties

Current literature on commonality predominately focuses on the benefits and penalties to the development and production phases of a system's life cycle. However, in order to more completely determine whether commonality should be implemented within a system, the effects of the benefits and penalties on all phases of the life cycle should be evaluated. The benefits and penalties are examined below as they apply to each phase of a system's life cycle.

In this section, it will be assumed that life cycle offsets exist between the two common systems. To simplify the discussion of the benefits and penalties, the first system developed will be referred to as system A, while the systems that are developed in the future are referred to as system B, although there may be a collection of future variants. Chapter 7 expands on these benefits and penalties by integrating them into an economic model to evaluate the opportunities for commonality.

2.4.1 Development

The development of common systems can either be conducted through forward commonality or reactive reuse. This section will describe some of the monetary and non-monetary benefits and penalties of commonality, and how they impact the system. Figure

6 demonstrates the monetary benefits and penalties on the development cost of system A and system B.

Benefits

- **Decrease in development cost and time** – The reuse of design and development work from system A, through either forward commonality or reactive reuse, decreases the amount of work that must be conducted for system B. The decrease in development work can take the form of lower costs or less time to market. The amount of time to market is an extremely important factor when competing with other firms and developers for market share (Robertson & Ulrich, 1998). This is a benefit to system B in both reuse and forward commonality.
- **Decrease in future system development risk -** Developing a future common variant requires a lower development investment, resulting in less development risk for system B. The lower development risk allows managers to develop a new common variant that may be considered to expensive or risky without the ability to use the common components (Gonzalez-Zugasti, Otto, & Whitcomb, 2007). The lower development risk also allows developers to create a family of products with greater system variability to meet the demands of more niche markets. This is a benefit to system B in both reuse and forward commonality.

Penalties

- **Cost of Common Development** – In order to develop forward commonality between system A and system B, additional development work must be conducted at the time that system A is developed. The increased development work is called the *cost of common development* and results because system A must be designed to meet the requirements of future systems in addition to those of system A. This is a penalty on system A in the case of forward commonality.
- **Integration Penalty** – When reusing development work from previous systems, the development cost associated with the common components is not completely eliminated because the common components must be integrated into system B.

The required work, referred to as the *integration penalty*, often takes the form of additional development and testing. This is a penalty on system B in both reuse and forward commonality.

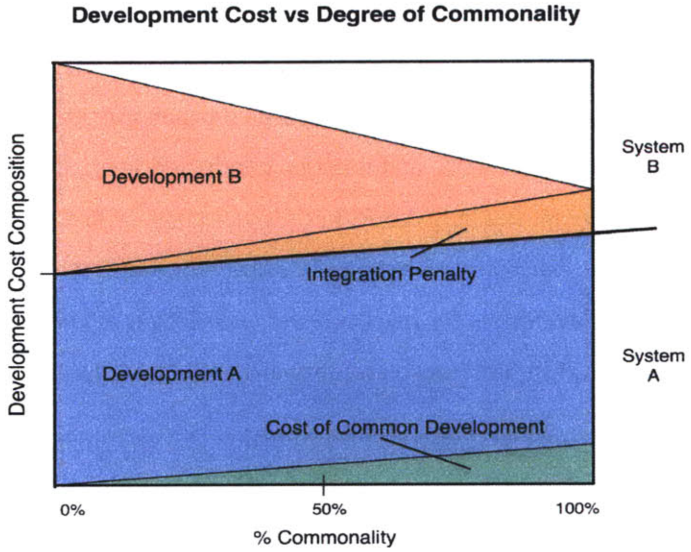


Figure 6: The categorization of development costs for system A and system B, as the percentage of commonality increases between the systems. The cost impact is based on the factors described above, including: decreased development work, *cost of common development*, and the *integration penalty*.

2.4.2 Production

In the following discussion of the effects of commonality, it is assumed that production costs are predominately composed of variable recurring costs. Figure 7 summarizes the economic effects of the benefits and penalties of commonality that are discussed in this section.

Benefits

- **Economies of scale** - Common components that are manufactured for several systems will be produced or procured in larger volumes than if commonality was not employed. Commonality causes firms to procure or produce larger volumes of the common components and, as a result, spread the fixed recurring costs across a greater volume of components, resulting in lower cost per item. The decreased cost per item is referred to as *economies of scale* (de Neufville, 1990). In the case of overlapped production, this benefit will affect the materials portion of the costs for both systems A and B in reuse and forward commonality.

- **Accelerated learning curve** – A learning curve refers to the fact that production is easier, cheaper, and faster the second time a product is produced. This holds true for each additional product until production reaches a limit in efficiency. An increase in the production quantity of common components leads to an *accelerated learning* curve (Robertson & Ulrich, 1998). In overlapped production, this benefits the labor portion of systems A and B equally, but as the offset increases the benefits to labor will be shifted to system B. The benefits are obtained in both reuse and forward commonality.

Penalties

- **Excess capability penalty** - Common components are often more complex than independently developed components because they are designed to meet additional requirements. The added complexity makes production more expensive compared to independent components. This penalty, referred to as the *excess capability penalty*, affects both systems but it may not affect them equally. In general, this penalty will affect system B in reuse and systems A and B in forward commonality.

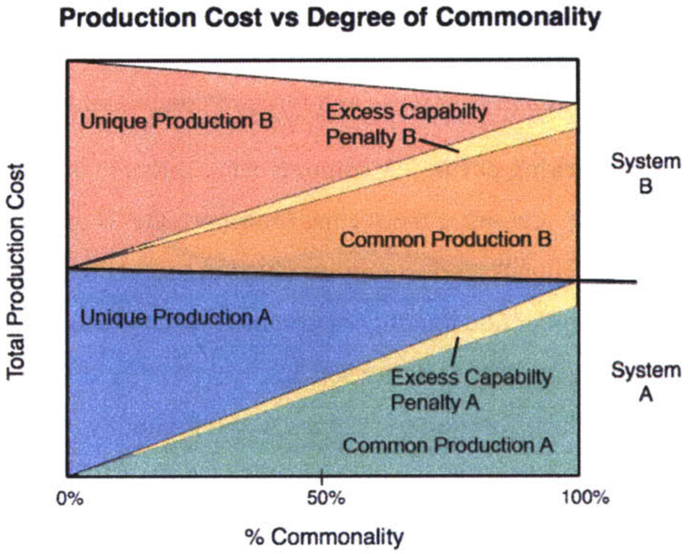


Figure 7: Production cost vs. percentage of commonality, based on *learning* curve benefits, *economies of scale*, and *excess capability penalty*; the plot shows that system A will likely receive marginal benefits, while system B receives greater benefits

2.4.3 Operations

Operations costs are composed of both fixed recurring and variable recurring costs. Fixed recurring costs are incurred whether a system is operated or not and is independent of the number of operations that occur. Variable recurring costs are the marginal costs per operation. Fixed and variable recurring costs are each benefitted in different ways: by a *capability overlap* and *an accelerated learning* curve, respectively.

In order to maintain the capability to operate a component there is a fixed recurring cost. If the component is common between multiple systems, the fixed recurring cost is shared between the systems, resulting in a benefit from the *capability overlap* to both systems or the later system, depending on the accounting method chosen.

The accelerated *learning curve* refers to the fact that completing a task is easier, cheaper, and faster the second time the task is conducted. Developing components and processes with commonality results in fewer unique components, and increased frequency in which the common component or process is operated. Commonality thus creates an accelerated learning curve in variable recurring costs as a result of the increased operation frequency.

The accelerated learning curve benefits variable recurring costs for systems A and B equally if operated in parallel, but as the offset increases the benefits will be shifted to system B. The benefits are obtained in both reuse and forward commonality.

The benefits to commonality, below, will be described as it applies to fixed recurring costs as a *capability overlap*, or to variable recurring costs as an *acceleration of learning*. Figure 8 represents the economic impacts of commonality on the operations phase for both systems.

Benefits

- **Decreased sustaining engineering** – Utilizing commonality will decrease the number of unique technologies and components, which in turn decreases the amount of required sustaining engineering. Sustaining engineering is the continued engineering and technical support of hardware that is in operation. NASA maintains technical expertise on hardware that is in operation in case that expertise is needed during an operation. If a common component or system is operated more frequently because it is used on multiple systems, the fixed recurring costs are not duplicated, but decrease because of the *capability overlap*. This is a benefit to both systems A and B, with an overlap in operations for both reuse and forward commonality.
- **Decreased logistics costs** – Logistics costs are the costs associated with storing, moving, maintaining, and securing hardware. Fewer unique components lead to smaller required inventories and lower logistics cost (Robertson & Ulrich, 1998) (Crites & Tremblay, 1989). The decrease in logistics costs results in a benefits to operations fixed recurring costs in the form of a *capability overlap* during years of parallel operations and to variable recurring costs from *accelerated learning*. This is a benefit to both systems A and B for reuse and forward commonality.
    - Decreased sparing requirements – Fewer unique components lead to fewer on-orbit spares required to maintain a constant risk profile (Siddiqi & de Weck, 2007)(Crites & Tremblay, 1989).

- **Decreased training costs** – Creating common components decreases the number of unique components, which in turn reduces the training infrastructure and the training requirements for crewmembers, support, and training personnel (Coan & Bell, 2006)(Crites & Tremblay, 1989). The decrease in training costs results in a benefit to fixed recurring operations costs in the form of a *capability overlap* during years of parallel operations and to variable recurring costs from the *accelerated learning*. This is a benefit to both systems A and B for reuse and forward commonality.
- **Decreased operations risk** - Reusing space proven equipment increases the confidence in the equipment and reduces the risk of equipment failure (Gonzalez-Zugasti, Otto, & Baker, 2000). Additionally, common operation interfaces reduce the complexity of operations, which also reduces risk of operator error (Coan & Bell, 2006). It is important to increase the ease of operations for human space flight because crewmembers are required to work in hostile and stressful situations. Operations within the aerospace industry in particular, are inherently risky because they deal with relatively new technologies that cannot easily be tested on Earth. Having space proven equipment greatly reduces the risk of operations. In fact, private launch companies must reuse a certain percentage of components in order to obtain insurance on launch vehicles and cargo.

Penalties

- **Excess capability penalty** - Common components are often more complex than independently designed components because they are designed to meet additional requirements. The added complexity makes operations more complex and expensive compared to independent components. This penalty, referred to as the *excess capability penalty*, affects fixed and variable recurring costs of both systems, but it may not affect them equally. In general, this penalty will affect system B in reuse, and systems A and B in forward commonality.

- **Decreased performance** - The application of commonality often includes a performance penalty because the component or system is less optimized for a particular operation.

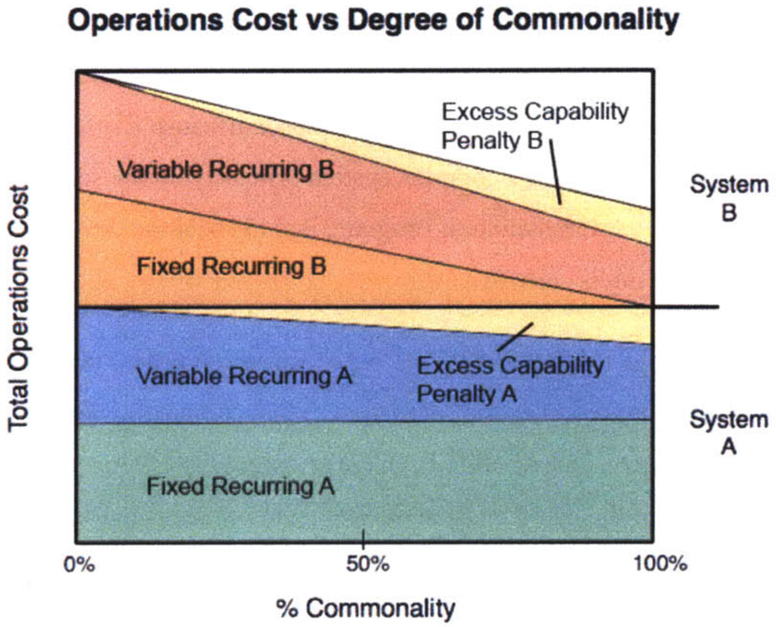


Figure 8: Operations cost vs. percentage of commonality. Benefits to fixed recurring costs occur only when the systems are operated at the same time. The chart shows all of the fixed recurring cost benefits accruing on system B, but the benefits can be shared by both systems, based on the accounting approach chosen.

2.5 Conclusion

This chapter presented the relevant introductory concepts and background on the management of commonality. The information presented, forms the basis of knowledge for the remainder of the thesis. The chapter includes a discussion of trends such as divergence and life cycle offsets, and how these trends affect the management of commonality, including a framework to classify the assets of a common system. Also, the chapter includes a description of the critical management tasks required to successfully manage forward commonality and the management approaches observed in the industry case studies. Finally, the chapter contains a detailed description of the benefits and penalties of commonality as described in literature.

3. Constellation Space Suit System

The first case study was conducted on the Constellation Space Suit System (CSSS). Each of the case studies was chosen because it offered a diverse environment in which commonality must be managed. The CSSS was chosen because it offers an example of (1) commonality applied within a complex system primarily composed of hardware, (2) it is part of the more recent Constellation Program, and (3) because development for the CSSS was contracted out by NASA.

The EVA System Project Office (ESPO) is the organization tasked to create the CSSS. They are specifically tasked to create a suit system that can protect the crewmember during (1) Launch, Entry, and Abort (LEA), (2) microgravity EVAs, and (3) planetary surface EVAs. Each of the listed mission environments creates competing requirements, prohibiting a single suit system. Despite the competing requirements, ESPO is working to minimize life cycle costs and the launch mass by creating a highly common and modular suit system. The details of the architecture requirements and resultant requirements are discussed in section 3.2, Constellation Space Suit Systems.

The primary method used by ESPO to manage commonality is to baseline all assets that could conceivably become common as common between the two suits and to subsequently control divergence to ensure that when divergence occurs it is for an acceptable reason. The detailed observations related to the management methods used, are discussed in section 3.3, Observations.

3.1 Background

In 2004, President George W. Bush gave NASA a new Vision for Space Exploration (VSE) that set NASA on a path to place people on the moon by 2020 and on Mars by the following decade (Bush, 2004). The new vision created a need for a completely new space transportation system because the current system, the Space Shuttle, can only be used for missions to Low-Earth Orbit (LEO). The new program was named the Constellation Program. This section outlines the architecture of the Constellation Program as it existed when the CSSS case study was conducted.

The architecture of the Constellation Program is divided into three major phases: (1) the initial capability (IC) phase, (2) the lunar capability phase, and (3) the Mars capability phase. Figure 9 shows the projects required for each of the phases. The initial capability phase, scheduled for launch by 2015, will involve launching up to four crewmembers into LEO in Orion using the Ares I launch vehicle. The IC phase requires that crewmembers are able to work efficiently in launch, entry, and abort (LEA) environments. The suits are required to return any consumables that it receives for ventilation, cooling, and breathing from the vehicle back to the vehicle, offering a highly closed-loop system.

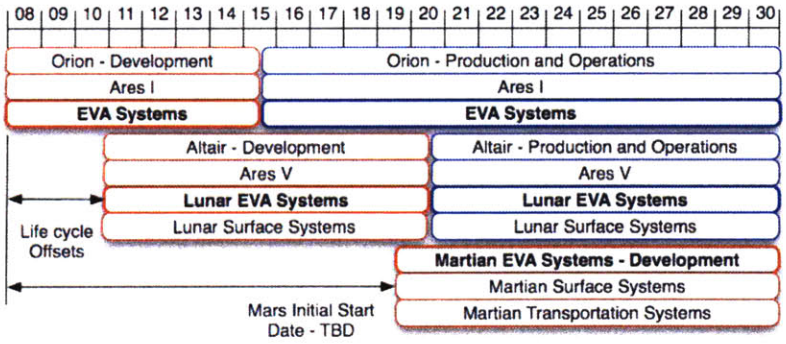


Figure 9: The Constellation Program schedule for the development (shown in red) and production and operations (shown in blue) for each phase (the development start date of the Mars Capability phase is not set, but operations are base-lined for 2030)

The lunar surface capability phase, scheduled for launch by 2020, will similarly launch the crew into LEO in Orion using Ares I, as seen in Figure 10. However, once in orbit the crew will dock with the lunar lander, named Altair, and the Earth Departure Stage (EDS). Altair and the EDS will be launched into orbit using the Ares V launch vehicle. The EDS will then be used to accelerate Orion and Altair towards the Moon. The crew will transfer to Altair, separate from Orion, and descend to the lunar surface. For extended stays, habitation modules will meet the crew on the lunar surface. Once tasks are completed, the lunar ascent module will return astronauts to the orbiting Orion vehicle. Orion will then be used to transfer the crewmembers back to Earth. The lunar capability phase requires the crew to operate during LEA, micro-gravity extravehicular activities (EVA), and planetary surface EVA mission environments.

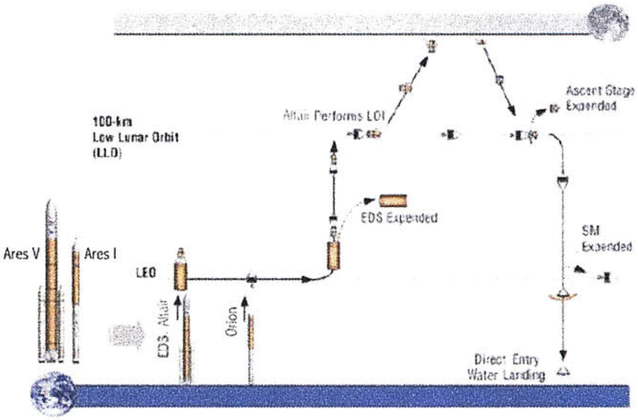


Figure 10: Constellation Program lunar capability phase architecture sequence, altered from: (ESAS, 2005)

The Mars capability phase was outlined in the Exploration Systems Architecture Study (ESAS) (Figure 11). Martian capability will additionally require the creation of a Mars transportation vehicle that will transfer the crew to and from Mars for a 500-day stay on the surface. The Mars capability phase requires that the EVA System support crew operations during LEA, micro-gravity EVA, and planetary surface EVA mission environments.

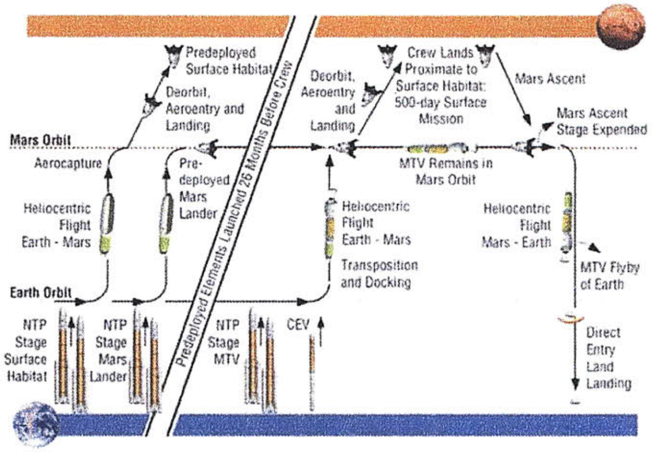


Figure 11: Current Martian Architecture, from (ESAS, 2005)

The Constellation Program is divided into seven projects based on the program's architecture. Each project is led by a project manager and supported by a number of groups, such as Systems Engineering and Integration (SE&I). Each project is then subdivided into elements; in the case of the EVA Systems Project Office (ESPO) they are: the suit element, the equipment and tools element, and the Vehicle Interface Element (VIE). Figure 12 shows the Constellation Program's hierarchy with regards to the EVA System Project.

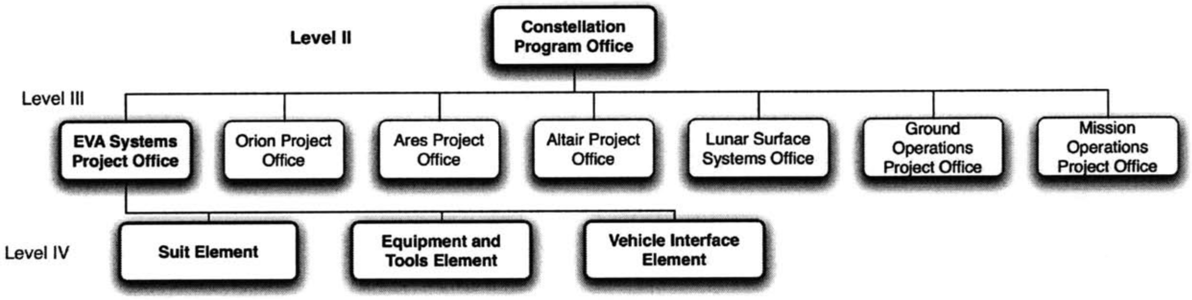


Figure 12: Constellation Program and EVA System project organization

3.2 Constellation Space Suit Systems

In addition to the new space transportation system, the Constellation Program also requires a new EVA system because the space suits currently in operation do not meet the requirements of the Constellation Program. The following section, Architecture Requirements, describes the technical requirements of the Constellation Program that affected the EVA System and how these requirements have influenced the CSSS architecture. The most significant impact is seen in the required operational environments: launch, entry, and abort, microgravity EVA, and planetary surface EVA. The resultant CSSS architecture will be composed of two common suit configurations. Details about the development and design of the architecture are discussed in section 3.2.2, Architecture Development. When the case study was conducted the acquisition process was in progress. Aspects of the acquisition that apply to the management of commonality are discussed in section 3.2.3.

3.2.1 Architecture Requirements

The EVA System Project Office (ESPO) is responsible for developing and maintaining the Constellation Program's space suits. Specifically, they are tasked to:

- Provide and maintain all equipment including pressure suits, EVA life support systems, umbilicals, EVA tools and mobility aids, EVA vehicle interfaces, EVA servicing equipment, suit avionics, individual crew survival equipment and ground support systems (NASA, CxP 70000)
- Protect the crewmembers and allow them to work effectively in the pressure and thermal environments that exceed the human capability during all mission environments, including:
    - Launch, entry, and abort (LEA)
    - Microgravity EVA
    - Planetary surface EVA
- Return consumables that it receives from vehicles back to the vehicle as a part of a closed loop system via an umbilical
- Implement commonality/interchangeability at the component and sub-system level should be applied to and across all systems and all missions of the exploration architecture where possible (NASA, CxP 70000)

The new EVA system will be operated in a very diverse set of mission environments: LEA, microgravity EVA, and planetary surface EVA. Missions on the Space Shuttle and International Space Station (ISS) only require space suits to operate in two of those mission environments: LEA and micro-gravity EVA. To meet the Space Shuttle and ISS environment requirements two independent suits are currently used. Figure 13 shows the suits currently in operation: the Advanced Crew Escape Suit (ACES) (left) which is used for LEA and the Shuttle Enhanced Extravehicular Mobility Unit (EMU) (right) which is used for micro-gravity EVA (Thomas & Mcmann, 2006).

The EVA System Project then is faced with two options: (1) develop a third suit designed specifically for planetary surface EVAs and redesign the current suits to meet the Constellation Program requirements or (2) develop a new EVA system that can meet the requirements of the various environments more efficiently. It is extremely inefficient to operate three independent suits, and as a result ESPO began to explore the development of a more efficient suit system that utilizes commonality between the suits.

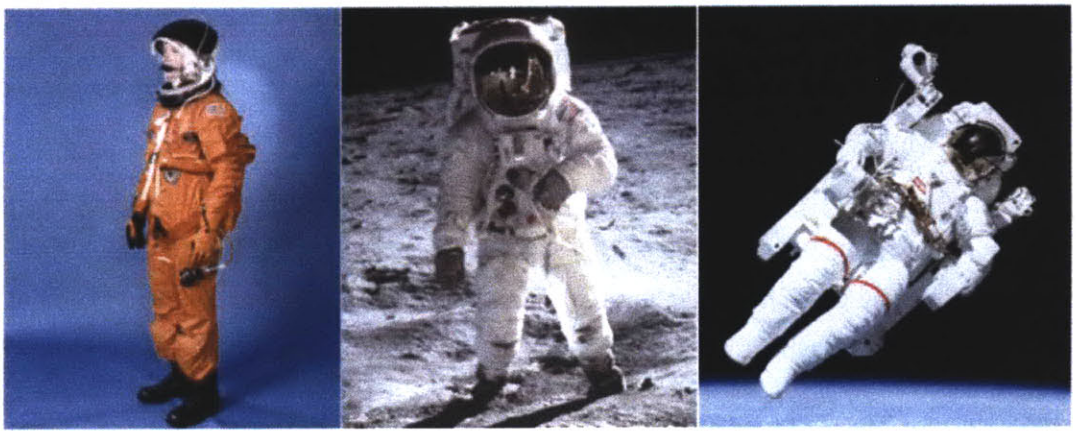


Figure 13: Images of the ACES (left), Apollo EMU (center), and Shuttle/ISS EMU (right), images from NASA

However, developing a new CSSS with commonality has proven to be a challenging task. Each of the mission environments in which the CSSS must operate lead to very different and often, competing requirements (Jordan, 2006). Figure 14 depicts some of the requirements for each of the mission environments that tend to compete with each other. The mission environments and associated design features are described in more detail below:

Launch, Entry, and Abort

During launch, entry, and abort (LEA) the space suit must support and protect the crewmember during seated operations, for unpressurized 1-G mobility (nominally the suit is pressurized no more than 1 psid), contingency crew survival operations, and protect against fire and toxic material contamination. The LEA suit currently in operation is the ACES (Figure 13, left), which was designed to be light in order to support 1-G unpressurized mobility and contains relatively few hard components. Hard components are avoided because they can lead to crewmember injuries in the case of off-nominal landings or a ballistic re-entry. The CSSS must also transport and return cooling and breathing gas from the vehicle during LEA operations.

Microgravity EVA

During microgravity EVAs the suit must support the crewmember for pressurized microgravity EVAs and protect them from thermal extremes and micrometeoroids. The need for pressurized mobility has led designers to use more hard components such as bearings and disconnects because they offer constant volume movement, requiring less energy to move. The most recent microgravity EVA suit is the enhanced EMU (Figure 13, right), which contains a hard upper torso (HUT), several hard bearings, and hard disconnects. The suit operates in microgravity in which it is virtually weightless, so the weight of hard components is less of an issue. During microgravity EVAs crewmembers receive breathing gas and cooling through either an umbilical (used for Mercury, Gemini, and Apollo) or a Portable Life Support System (PLSS)(used for Shuttle and ISS).

Planetary Surface EVA

The only space suit that has ever operated on a planetary surface is the Apollo Extravehicular Mobility Unit (Apollo EMU), which was used for a maximum of three EVAs per mission (Figure 13, center). Missions to planetary surfaces will require pressurized walking mobility, thermal protection, micrometeoroid and dust protection, and longevity, depending on the planned mission length. The Apollo EMU was composed of a combination of hard and soft components to minimize weight while still maximizing crew mobility. One area that was not focused on during the development of the Apollo suit was longevity and dust protection. The Apollo suits were used for a maximum of three surface EVAs, but at the completion of the mission, the connectors and bearings became stiff and difficult to operate. In addition, soft goods became visibly damaged and torn from dust abrasion. Dust protection is a great concern for future lunar missions.

The most cost-efficient solution to the varying operational requirements would be to create a single suit that operated in all environments, as was done in the Apollo program with the Apollo EMU. However, learning since the Apollo program has shown that the performance and safety compromises that are required in order to operate a single suit are too dangerous. One of the primary concerns is the suit's entry method. Current entry methods involve either a waist disconnect (EMU), rear hatch (Russian Orlan, MK-III, and ILC rear-entry I-Suit), or zipper (ACES and Apollo) (Thomas & Mcmann, 2006). The waist disconnects and rear hatch entail large hard components that could injure the crew during LEA. Zippers were used to minimize the risk of crew injury on the Apollo EMU, but the zippers were heavily contaminated with dust and became difficult to operate after just three EVAs. The poor performance has prohibited the use of zippers for future lunar surface suits.

The entry method, among other design barriers, makes it dangerous to use one suit. However, on the other end of the spectrum, three suits that are individually optimized for each mission environment would have the best performance, but it would be extremely costly to develop, manufacture, launch, and operate three suits. The trade then is between the cost benefits of one suit and performance benefits of multiple suits. The EVA System Project's solution is to create two common suits. The mission environments for the two common suits are represented in Figure 14, with the configuration 1 suit being developed for the initial capability phase and the configuration 2 suit being developed for the lunar capability phase.

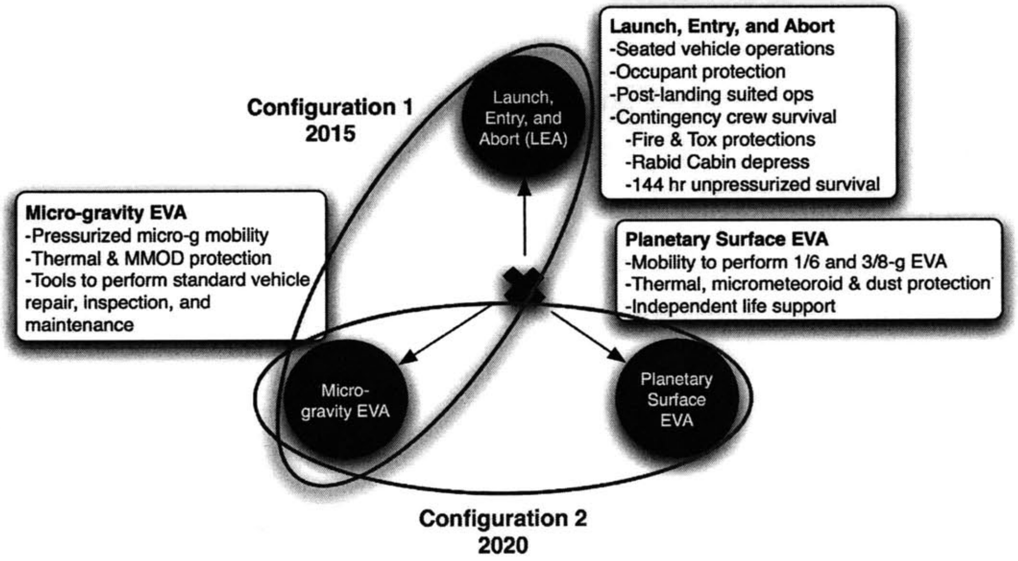


Figure 14: The requirements of the three mission environments are competing (adapted from NASA)

There are two additional factors that pushed the project towards the creation of two common suits. First, the project contains staged milestones: IC is required by 2015, lunar capability is required by 2020, and Mars capability by 2030 (Figure 9). Resources are more efficiently utilized by breaking the work into multiple development projects. NASA must work with limited financial and human resources and a staged development approach allows for a steady workload in which resources can be managed across the entire time span. Conversely, the development of a single suit that meets all environment requirements by the first milestone would front load the development work and be extremely costly. Second, the creation of two suits allows for a decrease in risk through learning. Lessons from the development, production, and operations of the initial capability suit can work to improve the lunar capability suit.

ESPO must still work to minimize life cycle costs and launch mass, while meeting the aggressive performance and schedule goals set by the program. Commonality between the two suits is one method that has been identified to accomplish these goals.

3.2.2 Architecture Development

The previous section discussed the technical design rationale behind the two common suit architecture. The following section will now discuss the development of the Constellation Space Suit System (CSSS) architecture and the management methods used to apply commonality, including: the development process and the management of the architecture's evolution over time. At the time that this case study was conducted the contract for the development for the space suit system had not been awarded. Despite the lack of a prime contractor, significant progress was made towards the development of the CSSS architecture. When the case study was conducted the acquisition of the suits was in progress, so details about the Request for Proposal (RFP) are discussed in section 3.2.3.

The development of the CSSS began with the development of the project's requirements. The EVA System Project Office (ESPO) created the project-level requirements by interpreting the program-level requirements and the Constellation Program (CxP) goals. The project-level requirements then flow down into the EVA System Requirements Document (SRD) and finally the element requirements documents (ERDs). Requirements are also added based on the operations concept for the system. All of the requirements documents then undergo several iterations of system reviews, concept developments, and industry Requests for Information (RFI). RFIs allow potential industry bidders to present feedback in order to mature requirements; industry is able to present more accurate bids if the requirements are mature. The requirements are continuously updated and matured as system concepts are further analyzed.

One of the requirements that had a significant influence on the architecture was a program-level requirement to implement commonality wherever possible (NASA, CxP 70000). This requirement, along with direct feedback from the program during the creation of the architecture, initiated a study to analyze each expected Contract End Item (CEI) in the architecture concepts and determine whether or not each CEI could feasibly be common between the suits. If the item could feasibly be common it was tagged as such in the baseline architecture, EVA System Reference 1 (ESR1).

Once the baseline architecture was created and documented in the EVA System Architecture Description Document (ESADD) any changes to the baseline must first be analyzed by the Architecture and Analysis Group, a part of the system-level SE&I group. If the Architecture and Analysis Group analysis determines that it is beneficial to the system then the proposed change will go through a multi-tier approval process involving control boards and systems engineering panels at the element-, project-, and possibly the program-level. It is a difficult process to make any changes to the architectural baseline, ensuring that if a change does happen it is for a beneficial reason.

The resultant ESR1 architecture that emerged from the described process is composed of two space suits with an extensive amount of common and modular components. The baseline architecture was so common, that it was described as two configurations of the same suit system: an initial capability (configuration 1) suit and lunar capability (configuration 2) suit (Figure 15). Modular, in this case is component-sharing modularity, in which the two suits share some modules for operations in order to offer launch mass benefits in addition to the benefits from commonality. For example, the exact same lower arm is used on both suits, but interchanged when the suit is operated. As a result of the modularity, the two connections for the lower arms are also the same on both suits. The

ESR1 suits contain modular arm assemblies; a modular integrated waist, hip, and leg; modular boots; and a modular helmet. The two systems are also composed of common umbilical interfaces, glove architectures, mobility rings, restraints, and a common Body Seal Closure (BSC). The suits also have unique soft upper torso's (SUT), visors, and thermal and micrometeorite garments (TMG).

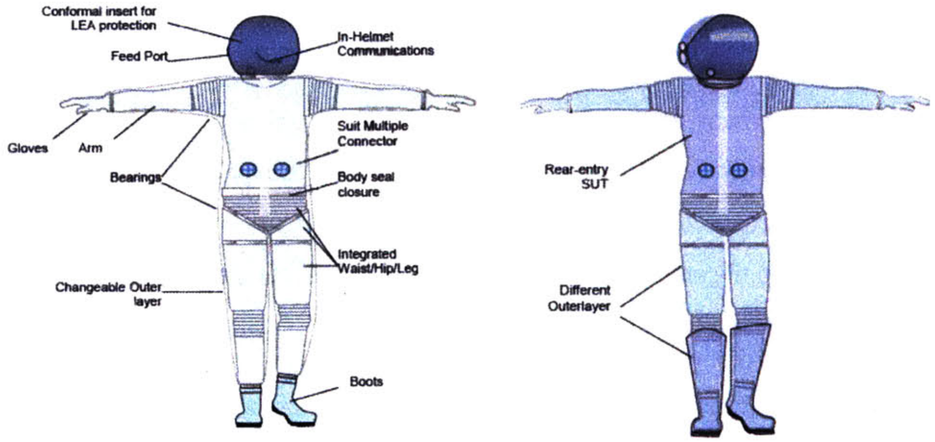


Figure 15: Depiction of ESR1 architecture, configuration 1 (left) and configuration 2 (right); areas of the same color are modular between the suits, with the only differences being the upper torso, visors, and thermal and micrometeorite garments, image from NASA

In 2008, the ESR1 architecture evolved into a new architecture for the pressure suits, EVA System Reference 2 (ESR2), shown in Figure 16. ESR2, however contains less commonality than ESR1, signifying that divergence has occurred. The new pressure suits are composed of modular lower arm assemblies, lower leg assemblies, helmets, and boots. It also contains common glove architectures, mobility rings, pressure garment restraints, and umbilical interfaces. However, the suits no longer have common upper arms, integrated waist, hip, and legs, and BSCs. The divergence or loss of commonality occurred as a result of learning.

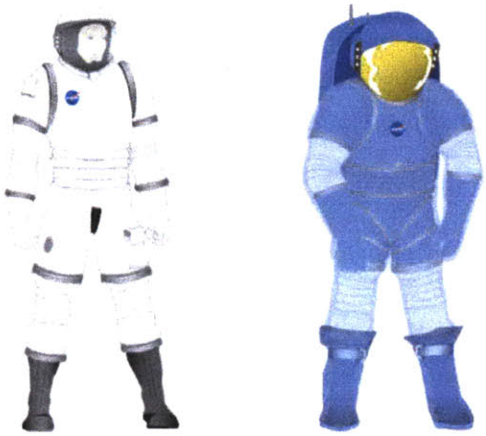


Figure 16: EVA System Reference 2 (ESR2) composed of configuration 1 (left) and configuration 2 (right), (NASA)

The proposed change that resulted in ESR2 was the result of a trade study to evaluate off-nominal landing injuries on crewmembers. The ESR1 architecture suits had common Body Seal Closures (BSCs) and a common integrated waist, hip, and leg assemblies. It was found that these elements, composed of several hard components, had the potential to cause back injuries in the case of an off-nominal landing. Other options were investigated that would allow the suits to remain common, including customized conformal seats and alternative suit and seat combinations, but those options could also be dangerous in emergency egress scenarios. The upper arm portion of the suit also diverged because of the different operations required. The configuration 1 suit was designed with a single scye bearing for unpressurized mobility, while the upper arm of configuration 2 was designed with a conformal fabric shoulder and shoulder bearing, which allows for a greater range of motion but is more massive.

A trade study was conducted between ESR1 and the proposed ESR2 and it was determined that the less common design was the better choice because there was less risk of crew injury. The trade study also showed that diverging from the common design could actually result in mass savings. Removing the BSC from both suits and replacing the hard lower torso components on the configuration 1 suit with a softer, less flexible design would result in lower system mass than the previous architecture.

Other Opportunities for Commonality

The following section discusses other opportunities for commonality within the EVA System. There are few functional similarities between pressure suits and other Constellation Program vehicles, but there are several other projects that are developing life support systems, presenting an opportunity for commonality with the Portable Life Support System (PLSS). The carbon dioxide and moisture removal technology for the PLSS, was chosen in part to take advantage of common technology with Orion and Altair. The baseline technology used for carbon dioxide and moisture removal is a Rapid Cycle Amine (RCA) System. The system utilizes amine beds that trap carbon dioxide and moisture. The amine beds can be rejuvenated with exposure to vacuum. Additional life support components, such as the variable pressure regulator, are also being investigated for use across multiple projects.

Another life support function, thermal control, is being developed in the CSSS with forward commonality in mind. The Spacesuit Water Membrane Evaporator (SWME) is being developed to operate as the thermal control system for the space suit. The SWME is being developed in large part to support commonality with the future Mars capability phase. The current system, the sublimator, was used effectively on the Moon during the Apollo program, and could be used for the lunar phase, but it cannot easily be used on Mars because the atmospheric pressure on much of the surface of Mars is above the triple point of water. The SWME is being developed because it is capable of working effectively in both environments and because it has the potential to be lighter than the sublimator, another factor that is more important for Mars (3/8 G) than the Moon (1/6 G).

Many of the elements are also working to maximize the internal commonality. The Portable Life Support Subsystem (PLSS) was able to reduce their part number count and thus logistics costs by creating common hose lengths, decreasing the number from 50+ lengths down to four in their baseline design.

Additional Observations

There are other factors that affect commonality within the CSSS, including the organizational structure, implementation of standards, and other Constellation Program requirements. These subjects are mentioned as additional observations.

The space suit system must interface with nearly all other projects. Integration between all of the projects can be a daunting task. To take on the task, ESPO created its own element, the vehicle interface element, to control all interfaces between the suit and the vehicles. This allows the element to work as the integrator, inherently promoting cross-project standardization and commonality in suit and vehicle interfaces.

ESPO also created the EVA Design and Construction (D&C) standards. EVA D&C standards require all components that are intended to interface with the suit to meet a common set of standards. These standards are meant to create a common and safe interface for EVA crewmembers.

Despite many efforts to promote commonality, one requirement created by the program is actually working to reduce the amount of commonality. Each project is required to develop and produce its own set of tools for the maintenance of its hardware. Therefore, Orion will create a set of tools used on Orion and Altair will also create a set of tools to be used on Altair, etc. The Constellation Architecture Requirements Document (CARD), however does state that projects are encouraged to create common tools with other systems. The rationale for this requirement was to make each project consider maintenance and tool masses in their design, however it creates a roadblock for commonality as the project that needs the tool first would have to pay the upfront cost of common development investment. The Orion project and EVA System Project were able to collaborate to create a common IVA/EVA camera to be used with the suit.

3.2.3 Procurement Process

NASA and other government organizations are presented with a unique challenge to manage extremely large development projects, much larger than most companies have the capital or resources to support. Within NASA, these development projects often involve high-risk and state-of-the-art technologies. In addition, NASA must manage these large, high-risk projects within the government procurement process and attempt to reconcile the contrasting goals between the customer (NASA) and the contractor to control costs. This section describes the challenges related to implementing commonality within the confines of the government procurement process as observed in the CSSS case study.

Industry's primary goal is to maximize profit for stockholders, while NASA's goal is to acquire the most value for their cost. NASA's value however is not completely monetary, but based on their system goals, usually related to technical performance and safety. The goals of the contractor and NASA are often competing, because most development contracts represent the fee as a function of total costs; cost-plus fee contracts. Cost-plus fee contracts give the contractor an incentive to increase the total cost in order to increase the total profit. ESPO made attempts to reconcile the goals of the customer and contractor by incentivizing the contractor to implement cost saving design techniques, such as commonality, in three ways: (1) the type of contract, (2) the award fee structure, and (3) the structure of the contract.

It is important to first understand the procurement process used by NASA. All government agencies, including NASA, must adhere to the Federal Acquisition Regulation (FAR), a set of rules that govern how the government acquires systems or services. The FAR was created to maintain public faith in the government and ensure that competitions are fair. In the acquisition process, government agencies first release a Request for Proposals (RFP) along with technical requirements documents. The RFP contains requirements for the project, a timeline to meet the project requirements, and outlines the contract and fee details. In the case of the Constellation Space Suit System, the RFP outlined a milestone-based cost-plus-award fee contract.

A cost-plus-award fee contract was chosen because it works to align the goals of the two parties by allowing them to share development risk. Figure 17 shows the amount of development risk that the contractor and owner inherit based on the contract type. At one of end of the spectrum, fixed price contracts place all of the risk. The higher risk on the contractor will cause proposals to be higher, to account for the development risk. A fixed price contract cannot be efficiently implemented unless there is relatively little risk and a defined quantity of work. At the other end of the spectrum, a cost-plus-percentage-fee contract places all of the risk on the customer and creates an incentive for the contractor to increase development cost, because higher development costs directly relate to higher fees. Cost-plus award fee was chosen for its balance and ability to incentivize cost control through the award fee structure.

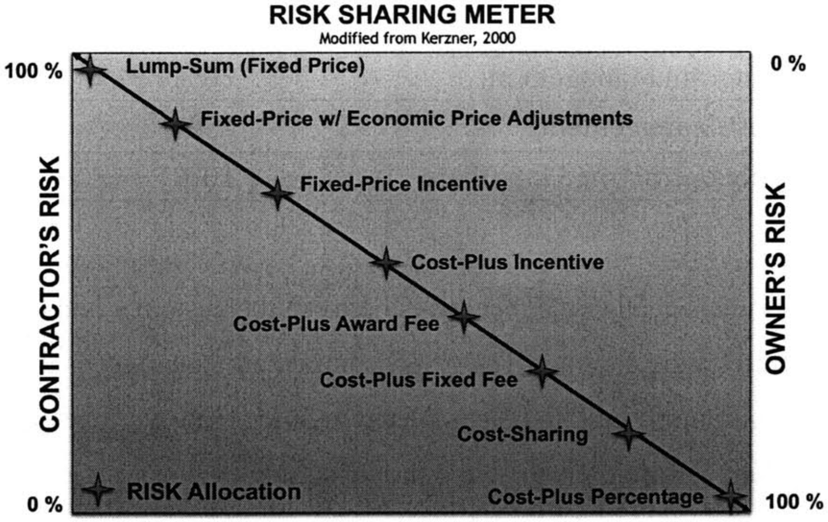


Figure 17: Diagram of risk sharing based on contractual structure

In the cost-plus-award fee contract described in the RFP (NASA, CSSS RFP) the contractor will be reimbursed for their expenses, however the award fee (a percentage of the total cost), which directly translates to the contractor's profit, is based on their performance in each milestone-based period. Each milestone involves a design review in which the contractor presents the current state and future plans of the project. The contractor must present the current design, the plan for commonality, the plan to create an open architecture, the interface control documents, the work efficiency index, and impacts to life cycle costs, among other subjects (NASA, CSSS RFP). A performance evaluation board then grades the contractor's performance for each milestone-based period. The performance is assessed on four categories: (1) technical performance, (2) program management, (3) cost management, and (4) subcontracting performance compared to goals. Each of the performance categories are weighted based on the project's priorities and can be changed at each performance period, but cost management cannot be weighted less than 25%. The approximate weights for each category are shown in Table 2. It is through this performance evaluation that ESPO is able to incentivize the contractor to control costs. The contractor is required to estimate the life cycle costs for the project and demonstrate the ability to control those life cycle costs.

Table 2: Award fee evaluation criteria and respective weights (NASA, ESPO, 2007)

Award Fee Evaluation Criteria	Weight
Technical	45%
Program Management	20%
Cost Management	25%
Subcontracting Plan Goals	10%

ESPO is also working to realign the incentives between the customer and contractor through the structure of the contract. The CSSS contract only guarantees the DDT&E work for the configuration 1 suit, to be completed in 2015. However, the contract has two built-in options. The first option is to award the sustaining engineering, maintenance, and production of the configuration 1 suits, through the year 2020; and the second option is to award the DDT&E of the lunar capability, configuration 2 suit. Each of these options will only be exercised if the project managers are satisfied with the work completed to date and the proposed costs. Creating the two tasks as options instead of a standard part of the contract is intended to encourage the contractor to meet cost and performance goals because ESPO has the ability to seek competitive bids.

However, there is a problem with the option structure, caused by a lack of competition in the space suit industry. Since, the first space suits were flown in 1963 there have only been a handful of EVA space suits developed: the Gemini IV IEVA Spacesuit (1965, David Clark Company), Gemini V-XII IEVA Spacesuit (1965, David Clark Company), Apollo/Soyuz Test/Skylab EMU (1968, PGS developed by ILC), and Shuttle/Enhanced Shuttle EMU (1981, PGS developed by ILC), the most recent of which was developed 27 years ago (although the Shuttle EMU was enhanced and flown in 2002) (Thomas & Mcmann, 2006). As a result, there is very little experience involving the design of a space suit system from the ground up and little competition for its development. Whichever contractor is awarded the development of the configuration 1 suit will likely be the only contractor that has the experience and capability to develop a cost-efficient configuration 2 spacesuit. However, ESPO is working to hold on to a feasible option to seek competitive bids by implementing open architecture design techniques and guarding against proprietary restrictions that were observed in the past.

3.3 Observations

The motivation for conducting the Constellation Space Suit Study (CSSS) case study was to identify the challenges and best practices in applying commonality within a NASA system. The CSSS case study presented information on the management of a recent system being developed within the Constellation Program, in which development is being contracted out, and of a system that is primarily composed of hardware.

The following section first discusses the challenges that were observed in the CSSS case study, followed by a detailed discussion of the observed methods used to manage commonality. The observed management methods are organized by the critical tasks for the management of commonality: managing the *identification* of commonality, *evaluating* opportunities for commonality, and the *implementation* of beneficial opportunities.

3.3.1 Challenges

Applying commonality to a set of systems has proven to be a difficult task (Boas, 2008). Each application has several challenges and tasks that must be overcome in order to obtain benefits from commonality. This section discusses the challenges observed in the CSSS case study.

Life Cycle Offsets

One of the key challenges for the CSSS was the existence of offsets between the development of the configuration 1 suit and the configuration 2 suit. The CSSS offsets are predominately created by the program's staging of milestones, i.e. operation of initial capability by 2015, lunar capability by 2020, and Mars capability by 2030. The program milestones are staged predominately as a result of resource constraints, but also to allow for learning; the missions to the Moon are meant to prepare developers and crewmembers for riskier missions to Mars.

The life cycle offsets have several implications for the management of commonality. The following challenges were specifically observed in the CSSS case study:

- Offsets have created an upfront penalty on the first system while offsetting benefits to the future. This upfront penalty can be of great concern especially when managers are working under a fixed budget and an aggressive schedule.
- Development offsets also create uncertainty in the future system, limiting the ability of engineers to identify opportunities for commonality with the future systems. Future uncertainty is a growing problem, as demonstrated by the administrations review of human space flight and additional program architecture options in consideration.

Divergence

Although the development of the CSSS is still in the architecting phase, divergence has occurred as demonstrated by the changes between the latest suit architecture, ESR2 (Figure 16), and the initial architecture, ESR1 (Figure 15). Divergence in this case was a result of learning. Impact tests done with prototypes of the original suit architecture showed that there was a substantial risk in having a Body Seal Closure (BSC) and other hard components on a suit used for Launch, Entry, and Abort (LEA). The CSSS is still under development and more divergence is expected to occur, so the challenge remains to ensure that when divergence does occur it is to benefit the system.

Procurement

While the seven industry case studies of commonality (Boas, 2008) consistently involved the in-house development of common systems, NASA is often faced with the additional challenge of managing commonality with contracted development. The challenge lies in the fact that the goals of NASA and the contractor are often unaligned. Contracting companies are attempting to maximize profit for stockholders, while NASA is attempting to obtain the most cost-efficient system. In most cost-plus fee contracts, contracting companies do not have an incentive to control costs, and in some cases they actually have an incentive to escalate costs.

3.3.2 Management Methods

This section discusses the observed management methods from the CSSS case study. The section is organized to explain the management methods used as they relate to the three tasks for applying commonality successfully: managing the *identification* of opportunities, *evaluating* opportunities, and *implementing* those opportunities that are deemed beneficial.

Manage the Identification of Opportunities

Identifying opportunities for commonality is the first task to manage when applying commonality to a system. Several factors were observed to be important in the identification of commonality, including the timing in which commonality is sought out and the method for which the opportunity was identified. Opportunities for commonality within the CSSS were predominately identified through active identification methods. However, there were also a few cross-project opportunities for commonality that were identified by passive identification methods. Active identification involves actions specifically made to identify commonality, while passive identification enables engineers to identify opportunities but are not necessarily made for the purpose of identifying commonality.

Timing

Observation: Commonality opportunities were identified early in the CSSS timeline, before the initial architecture was released.

The initial identification of commonality opportunities took place early in the development cycle and before the creation of the initial architecture. It was important for

ESPO to identify opportunities early in development because once an architecture is baselined NASA's change control process makes it difficult to change the baseline. All changes involve lengthy trade studies and a multi-tier approval process. Conducting trade studies and passing a change through the approval process takes time and money, which means that as time progresses the possible benefits received from identifying a new opportunity for commonality and the ability to influence costs decays. This concept is displayed in Figure 18 (Schulz, Clausing, Negele, & Fricke, 1999).

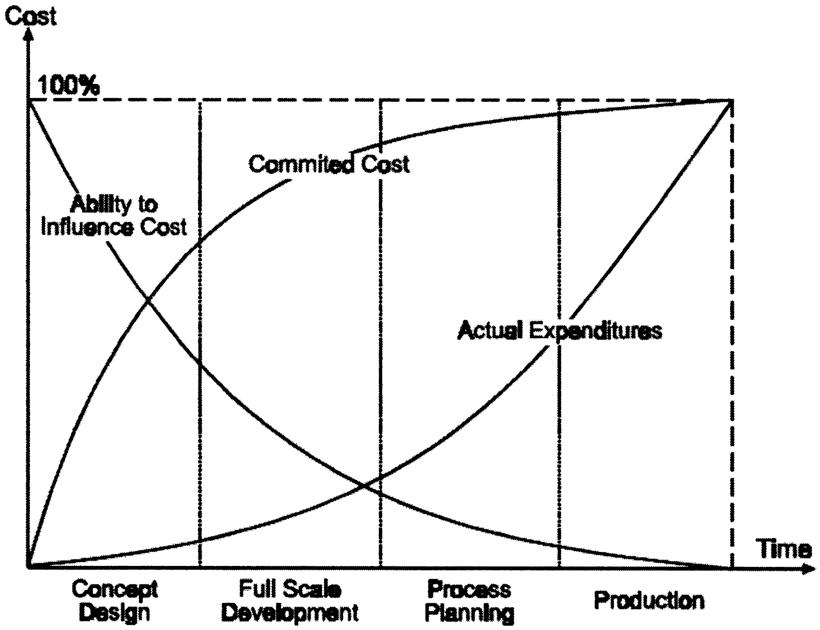


Figure 18: Model of committed funds and the ability to influence cost in a project's life cycle(Schulz, Clausing, Negele, & Fricke, 1999)

Active Identification Methods

Active identification methods are the management methods used for the specific purpose of identifying opportunities for commonality. In this case study they include: management support for commonality, assigning responsibility for commonality, and technical identification methods.

Management Support

Observation: The effort placed on the identification of common components is directly related to the amount of management or program-level support.

From the beginning of the project, there was a substantial amount of management or programmatic support encouraging commonality. This support can easily be seen in the Constellation Architecture Requirements Document (CARD):

> *"Commonality/Interchangeability at the component and sub-system level should be applied to and across all systems and all missions of the exploration architecture where possible… "* (NASA, CxP 70000)

In addition to documented requirements, ESPO also received direct feedback during the creation of the initial architecture that encouraged stakeholders to minimize the number of unique components to as few as possible. This encouragement sparked an increased effort to investigate opportunities for commonality. As a result, stakeholders evaluated each expected Contract End Item (CEI) to determine whether or not it could be made common between the configuration 1 and configuration 2 suits. The resultant architecture was created with a substantial amount of commonality and modularity.

Commonality requires an up-front investment that can be difficult to fit into a project's budget. In most cases a project will not willingly place itself at risk of going over budget to implement commonality. Program-level support for commonality gives the project the necessary stimulant to invest in commonality. The importance of program-level policies and support was confirmed in a study of NASA ground systems commonality, conducted by Quinn (2008).

Assigning Responsibility

Observation: Commonality was not holistically sought out until responsibility for doing so was placed on a group of stakeholders.

As stated above, the program requirements documents specify that commonality should be implemented wherever possible. However, the CARD does not state how the application of commonality will be verified or who is responsible for identifying opportunities for commonality. This leaves the responsibility of enforcing the application of commonality up to the project. In the case of the CSSS the task of identifying commonality was assigned to the stakeholders creating the initial space suit architecture.

After the baseline architecture was created, responsibility for evaluating architecture changes and commonality was placed in the hands of the Architecture and Analysis Group within the project-level SE&I group.

Responsibility for developing commonality within the system is also assigned to the contractor in the CSSS RFP. The RFP explicitly assigns responsibility for the identification of opportunities for commonality to the contractor. At each milestone the contractor must present the commonality plan and the effects that this plan will have on life cycle costs (NASA, CSSS RFP).

It is unlikely that individuals will invest the time to identify commonality unless it is their responsibility to do so. NASA is constantly forced to do more with fewer resources as a result of increasingly high expectations with an often shrinking budget, leaving projects and employees stretched thin. If the task of identifying commonality is not made an individual or group's responsibility, it can easily be overlooked.

Technical Identification Methods

ESPO identified commonality by analyzing the requirements, the technology choices, and operation phases for each expected CEI to determine whether or not the systems could become common. This process seems to have been successful, but the current architecture is expected to diverge. It is difficult to say whether or not a different approach would have been more efficient on the CSSS, but the method used by ESPO may be difficult to apply to larger systems because it would be increasingly difficult to analyze each component or end item.[1]

Passive Identification Methods

Passive identification methods are the management methods that are not used to specifically identify opportunities for commonality, however they increase the likelihood that commonality will be identified.

---

[1] There are several other systematic approaches to identifying commonality within a system, more information can be found in the referenced sources: Hofstetter, W. (2009). Stone, R. B., Wood, K. L., & Crawford, R. H. (2000). Dahmus, J. B., Gonzalez-Zugasti, J. P., & Otto, K. N. (2001). Weiss, D. M. (1998). Hofstetter, W., de Weck, O., & Crawley, E. (2005).

Increasing Visibility

Observation: Knowledgeable engineers that had visibility across the system and project were able to identify opportunities for commonality.

During the investigation it was found that in a few specific cases knowledgeable engineers were able to identify opportunities for commonality because they had visibility across the system and into other systems. Examples include common hose diameters between the PLSS and umbilicals, common variable pressure regulators between the PLSS and the life support systems on the Orion and Altair vehicles, and common EVA/IVA helmet cameras. This visibility was created through several different means, such as the organizational structure, integration groups, and other knowledge sharing methods. Increasing the visibility of engineers will increase the likelihood that an opportunity for commonality is seen. Visibility has been noted as an important factor in other commonality studies as well, such as a study of International Space Station commonality conducted by Waiss (1987).

Common Engineers

Observation: The same NASA engineering team will be used for the development of the configuration 1 and configuration 2 space suits.

ESPO is planning to use the same development team for both the configuration 1 and configuration 2 space suits, by gradually transitioning them as the projects stage over. Using the same engineers on common systems greatly increases visibility because engineers have an incentive to consider the future system while making design decisions for the first system. This allows the future system to be inherently represented in decisions. Additionally, it allows the most experienced team to develop the future system; after the development of the configuration 1 suit no other team is more qualified or knowledgeable to develop the configuration 2 suit. Using the same development team is not always possible if projects have highly overlapping or even parallel development schedules and insufficient resources for supporting the required development.

The same contractor team may also be used to develop both the configuration 1 and configuration 2 suits, because the contract is structured to include an option for the

DDT&E of the configuration 2 suit. The development option encourages the contractor to consider the future system during development of the first.

Organizational Structure

Observation: NASA has a crosscutting, function-based organizational structure that increases visibility across the program.

Employees within NASA's Constellation Program are organized into two separate structures, the Constellation Program structure and the center management structure. The center management structure organizes employees based on their technical expertise or function, while the program structure organizes employees based on the project or system that they are working on. Figure 19 depicts the center and program organizational structures and how these structures create a crosscutting organizational structure. For example, within the Engineering Directorate, there exists a Crew and Thermal Systems Division (CTSD) in which all engineers work on systems related to crew and thermal systems. The function-based grouping makes it likely that within one group or division many members could be working on similar functioning systems for diffierent projects. This type of grouping can lead to the identification of commonality because opportunities for commonality often exist between systems that perform the same function. Within the CSSS there were multiple cases in which employees learned of an opportunity for cross-project commonality by talking to group and office mates working on similar projects. One example was indicated by a life support engineer developing the variable pressure regulator for the PLSS that identified the opportunity for commonality with the Orion and Altair vehicles.

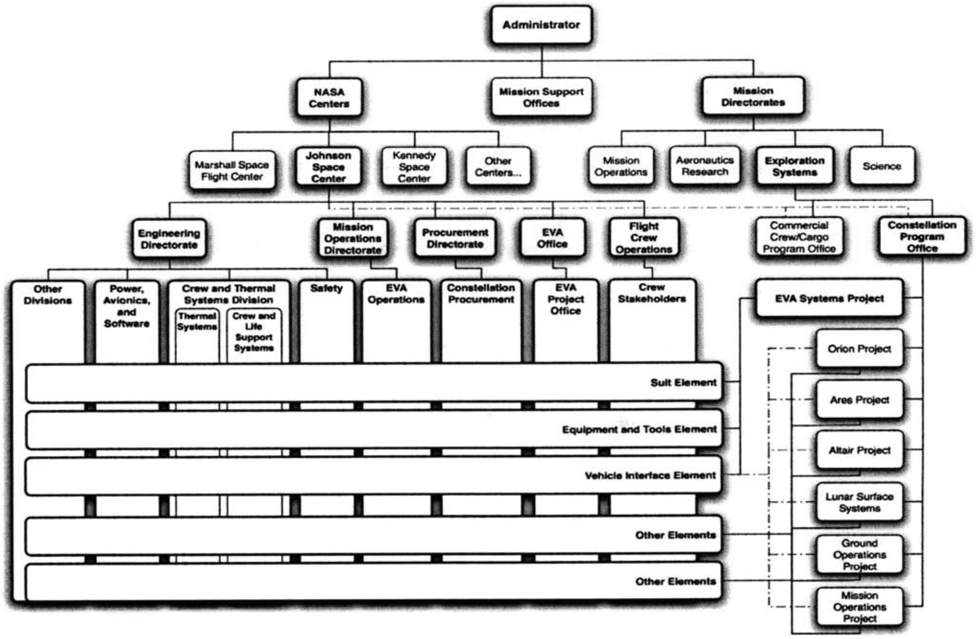


Figure 19: Organizational Structure of the EVA System Project

Integration Groups

Observation: Integration groups are used extensively at NASA to share information between projects and elements.

There are several integration and working groups within NASA that were created specifically for the purpose of sharing information between projects and elements. The integration groups were not created to identify commonality, but more generally to communicate and share knowledge between different groups. However, as they share knowledge they are effectively increasing visibility across projects and enabling engineers to identify commonality. Two integration groups are used extensively across the program, SE&I and SIGs. SE&I is a multi-level integration office that works within elements, projects, and the program. System Integration Groups (SIGs) also work to integrate projects but alternatively are function-based organizations. The SIGs work across all projects in order to integrate function-based knowledge. Categories of SIGs include: thermal/ECLSS, EVA, flight performance, structural loads, environments, and humans. Figure 20 shows how the SE&I and SIG integration groups fit into the Constellation Program organizational structure.

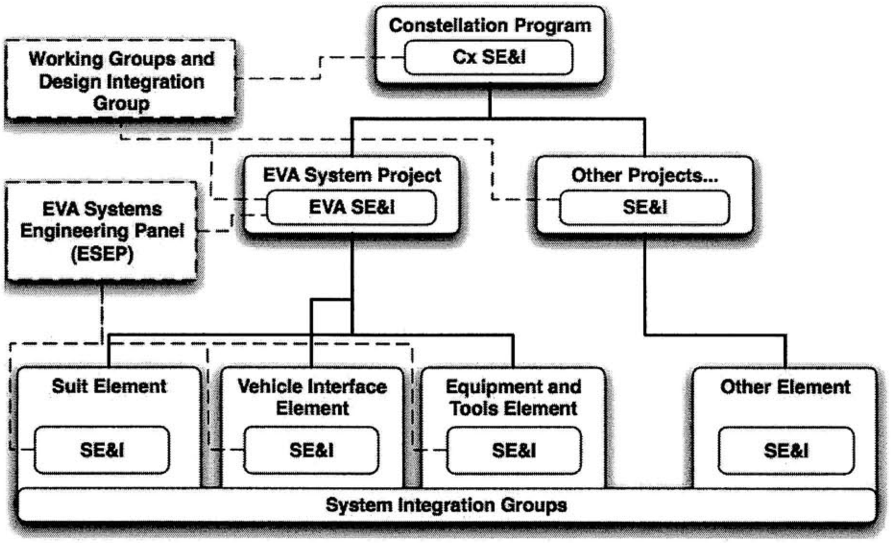


Figure 20: Integration groups and their involvement, in which dashed lines represent lines of communication and solid lines represent managerial authority

Knowledge Sharing

Observation: Knowledge sharing multimedia is used to inform individuals about other systems across the program and organization.

NASA has an extensive multimedia network that is designed to distribute information throughout the administration. NASA takes advantage of this multimedia network to increase visibility for engineers. This network includes *NASA Spinoffs*, *Academy Sharing Knowledge (ASK), JSC Roundup*, and others. Each of which have articles focusing on new development projects and technologies in use.

Incentivize Cost Management

Observation: The RFP documents released for the procurement process uniquely address and incentivize cost management and commonality.

Contracting companies and NASA have very different incentives. ESPO's goal is to purchase a cost-efficient space suit system that meets all requirements within an outlined schedule, while the contractor's goal is to maximize profits for their stockholders. The Request For Proposals (RFP) released for the Constellation Space Suit System (CSSS) outlines a milestone-based cost-plus-award fee payment method that is designed to reconcile the differing goals (NASA, ESPO, 2007). The contract incentivizes cost management and commonality in the contract type, the award structure, and the contract structure.

A cost-plus award-fee contract was chosen because it creates a balance of contractor and customer risk. In this award-fee payment method, the fee is based on a performance evaluation of four categories, as seen in Table 2: technical performance, program management, cost management, and performance of subcontracting goals. The weights of each category can change as the goals of the project change, but the cost management category must be weighted a minimum of 25%. The performance-based award-fee creates an incentive for the contractor to control life cycle costs, and as a result make an additional effort to adhere to the project's cost saving strategy of commonality.

The contract structure was also uniquely structured to incentivize cost savings. The RFP outlines options for the development of the configuration 2 suit and the production and operation of the configuration 1 suit. The options allow ESPO to avoid the competitive bid process if they are satisfied with the proposed costs and work completed of the current contractor, but the contract also gives ESPO the flexibility to seek competitive bids if they are unsatisfied. Presenting the development of configuration 2 as an option instead of a part of the contract creates an incentive for the contractor to meet performance objectives. However, the use of options instead of apart of the contract, creates some risk of encouraging the contractor to be single-product focused.

Evaluate Opportunities

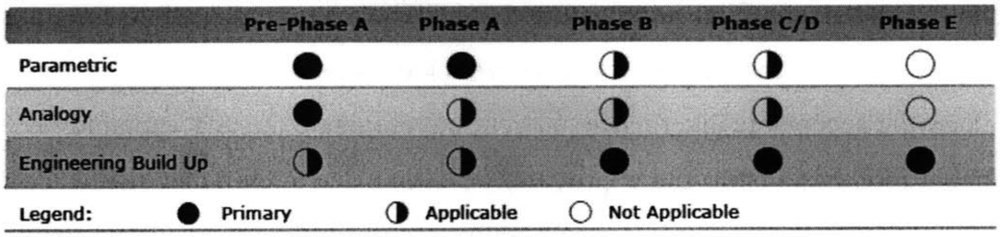
Commonality does not always benefit a system, but often implies a trade. The second task observed in managing commonality is the evaluation of opportunities to ensure that the opportunity is beneficial. This section looks at how the benefits and penalties of commonality were evaluated before the decision was made to implement.

Current Evaluation Method

Observation: NASA's current evaluation method does not specifically consider all life cycle benefits and penalties of commonality.

Within the EVA System, the Architecture and Analysis group evaluates each opportunity for commonality and each potential change or divergence from the baseline by conducting trade studies. Each trade study evaluates the design decision based on costs, performance, and safety, along with other technical performance measures. However, the economic portion of the evaluation does not take into consideration the specific benefits and penalties of commonality. Instead, the qualitative economic benefits of commonality are evaluated by comparing two designs and assuming that a common part would cost less. For an experienced engineer who is intimately familiar with a particular component or design, evaluating the development cost benefits for a component may be possible. However, evaluating the full life cycle cost benefits and penalties, including the effects on manufacturing and operations phases for a large complex system is much more difficult.

The problem lies in the fact that no method was found to quantitatively estimate life cycle costs of systems with commonality. NASA uses three methods for estimating a system's future cost: parametric cost models, analogy, and engineering build-up (2008 NASA Cost Estimating Handbook). Parametric cost models, such as NAFCOM, estimate development costs by using information from past NASA and Air Force projects. Parametric cost models are created using regression analysis to identify parameters, such as volume and mass to estimate the development and first unit cost. Cost estimating by analogy, similarly, is a way of measuring cost based on the cost of a past similar project. For example, if a previous helmet cost $10K to develop, than a new one will cost about the same depending on the degree of similarity. Engineering build-up is a way of estimating costs by getting a price quote on the expected CEI. Parametric models and cost estimating by analogy do not explicitly consider commonality when estimating costs, and engineering build-up cannot be used on large complex systems in the architecting phase, as seen in Figure 21, at the time that commonality should be identified and evaluated.



	Pre-Phase A	Phase A	Phase B	Phase C/D	Phase E
Parametric	●	●	◐	◐	○
Analogy	●	◐	◐	◐	○
Engineering Build Up	◐	◐	●	●	●

Legend: ● Primary ◐ Applicable ○ Not Applicable

Figure 21: Cost estimating methods and time of their applicability (2008 NASA Cost Estimating Handbook)

The RFP for the CSSS contract requires the contractor to estimate life cycle costs and utilize commonality to control costs, but the current situation may create a "fox guarding the hen house" scenario. The contractor is paid based on their own model's projection, which may lead them to create a model that is biased. Costs may be controlled more efficiently if an independent party produced the life cycle cost estimates.

To further investigate the economic evaluation of the benefits and penalties of commonality, an economic model was developed that specifically considered the benefits and penalties of commonality. The economic model is discussed in Chapter 7, utilizing information from the CSSS case study in order to discuss the evaluation method.

Implement Beneficial Opportunities

The last task for the management of commonality is to successfully implement the opportunities that are deemed beneficial. There are several factors, previously identified, that cause the implementation of commonality to be difficult, including divergence. The methods that were observed in the course of the CSSS case study are discussed in this section.

Manage Divergence

Observation: ESPO has a multi-tier evaluation and approval process that effectively reduces the less acceptable causes for divergence.

Divergence occurs, for acceptable reasons and sometimes for unacceptable reasons (Boas, 2008). The goal is to allow acceptable causes for divergence to occur while preventing the unacceptable causes. ESPO planned for divergence, and as a result intentionally created a baseline architecture that had the maximum feasible amount of commonality, ensuring that all opportunities were identified initially. The goal thereafter was to allow the acceptable divergence to occur but eliminate the unacceptable divergence. ESPO has a multi-tier approval process in place to eliminate the unacceptable causes of divergence. Each change to the baseline architecture requires a trade study and approval from the element control board, the EVA System Engineering Panel (ESEP), the Project Control Board (PCB), and the program if the change affects any other projects.

Decreasing the Integration Penalty

The integration penalty is the cost of implementing a previously developed component into a future system. The penalty may be associated with the cost to test or certify the system for production and operations. If the integration penalty becomes too high it could offset the benefits from commonality and encourage divergence. Implementing design for flexibility and open architecture techniques can reduce the integration penalty.

Design for Flexibility

Observation: Design practices that create additional flexibility in the system can work to preserve potential commonality.

Design for flexibility is to design a system so that it can be cheaply changed or upgraded in the future. The same techniques used to allow a system to evolve can also be used to reduce the integration penalty on future common systems. Design for flexibility also increases the likelihood that reuse and forward commonality will be successful. This increased probability of success exists because the flexibility allows engineers to more easily overcome design problems that could otherwise cause divergence. For example, if forward commonality is designed into a connector, but it was found that the design was insufficient for the second application, additional flexibility may allow engineers to cost-effectively make changes to either system while maintaining commonality. Modularity is one method of making a system more flexible (Fricke & Schulz, 2005).

NASA incorporated changeability into the Constellation Program requirements documentation; the CARD states the following (NASA, CxP 70000):

> *Growth potential includes both the capability to support evolving mission requirements as well as the capability to support technology upgrades throughout an Architecture system's design life. As technology evolves, there will be potential both for growth in capability and for compatibility issues between newer and older systems. Design decisions in areas where technology is rapidly evolving (e.g. electronics/avionics) should minimize the complexity required to perform future upgrades. Technology upgrade decisions will, in some cases, be driven by the benefit (e.g. lower life cycle cost, increased reliability) associated with an upgrade while in other cases, upgrades will be driven by the need of existing Architecture systems to interface with new systems that are developed years later with significantly more advanced technologies.*

Open Architecture

Observation: Certification costs can be extremely high and limit future changes or upgrades to the system. An open architecture approach is used to address the certification costs and to give NASA the flexibility to seek alternative developers.

One of the cost hot spots associated with the integration penalty is the certification cost. Designing a system with an open architecture approach requires that each CEI has its own Interface Control Document (ICD) and is individually certified. The ICD defines the interfaces of the CEI to the extent that it can be replaced with another component as long as the component meets all of the ICD requirements. Designing with an open architecture approach eliminates the need to recertify the entire system for each design change, thus reducing the integration penalty of future systems. An open architecture approach also enables NASA to seek alternative developers for future changes or upgrades without the trouble of recertifying the entire system The open architecture approach is further elaborated in the RFP (NASA, ESPO, 2007):

> *The Contractor shall use a modular/open systems approach in the design of the CSSS hardware. All Contract End Items (CEIs) and associated interfaces shall be separately certified. The Contractor shall develop and maintain ICDs for all interfaces between CEIs in accordance with DRD CSSS-T-003, Interface Control Documents. The Contractor shall use an approach such that any change at the subassembly level or below which does not affect the subassembly ICD, does not require recertification of the assembly, subsystem, element, or system.*

Implementing Standards

Observation: The EVA Design and Construction Standards have been implemented to control EVA interfaces.

In the past, several different groups have independently designed equipment in which the crew interfaces. Increased crew interface variability creates more complex operations. Increasingly complex operations can increase risk of operations and increase the overall training load (Coan & Bell, 2006). The EVA System created the EVA Design and Construction (D&C) standards to reduce crew interface variances. D&C standards require that all equipment that the crew will interface with must meet a certain set of standards for ergonomics, safety, and simplicity.

Creating Traceability

Observation: Design rationale is tracked and recorded in the EVA Systems Architecture Description Document (ESADD).

There have been a number of accidents in the past because commonality was implemented without complete understanding of the implications. The Ariane V first test flight failure was contributed to the reuse of software from the Ariane IV launch vehicle. Similarly, the Mars Climate Orbiter failure was contributed to reuse (Leveson & Weiss, 2004). Although individual components are highly reliable, the entire system may still be unsafe as a result of interaction errors. Commonality complicates this matter because common components are not designed and optimized for a specific application. ESPO is making an effort to track all of the intended design behaviors and design rationale of the architecture in order to pass on knowledge of the system. The hope is that when the system is reused in a new environment the architecture and the behaviors of the system will be well understood enough that interactions can be more accurately predicted. ESPO is recording design rationale with each architecture change in the EVA Systems Architecture Description Document (ESADD).

Addressing Proprietary Restrictions

Observation: Commonality has been inhibited in the past because of proprietary restrictions.

Many of the engineers working on the space suit system noted that when attempts to reuse particular components were made in the past, they were unable to because of proprietary restrictions. In one particular case a contractor created a particularly valuable EMI backings that was extremely useful within space suits. However, when another suit was being developed in the future, the component could not be used because they were proprietary to the original contractor. The engineers were forced to use an alternative component.

There have also been cases in which NASA has attempted to switch contractors, but were unable to find a competitive alternative because the original contractor claimed that a large amount of the Ground Support Equipment (GSE) was developed on internal funds and as a result proprietary. ESPO is working to limit these proprietary restrictions through intelligent contract writing. The RFP requires the contractor to hand over designs for all systems and components related to producing or supporting the CSSS. This is an essential portion of the contract if the government wants to hold onto the option to seek competitive bids on the CSSS. The RFP states that the contractor is required:

> *To provide engineering data defining the design to the extent required to support system design, manufacturing, test, and logistics support of the CSSS hardware and required spare parts. Engineering drawings and associated lists shall be sufficient to depict the detailed configuration of all system, subsystem, and component levels and to include ground support equipment (GSE). This DRD applies to released manufacturing, layout, assembly and installation drawings and schematics for all CSSS hardware. The level of detail presented in the drawings shall progress from the system, element, and sub-system level to the CEI and CI level over the course of the hardware design life cycle. All drawings used by the Contractor and their sub-tiers shall be delivered to NASA. The aggregate set of drawings shall define every dimension of the CSSS hardware* (NASA, ESPO, 2007).

3.4 Conclusion

Commonality can greatly benefit a system if it is managed properly. A case study of the Constellation Space Suit System (CSSS) was conducted to determine how commonality is currently being managed and identify the best practices and key challenges for applying commonality. This chapter outlines the first of three NASA case studies that are used collectively, along with other literature on commonality, to create managerial guidance for the application of commonality.

The application of commonality within the CSSS began with strong managerial support for commonality, by requiring commonality to be sought out in the CARD and giving direct feedback to the system architects encouraging the identification of commonality. The managerial support initiated a study of each expected Contract End Item (CEI) to determine if the CEI could feasibly become common, if it could it was baselined as such. The process used by ESPO may be difficult to repeat on larger projects, so other technical identification methods should be investigated.

To evaluate opportunities for commonality the Architecture and Analysis group conducts case studies for each proposed change from the initial architecture. The evaluation process used to determine if an opportunities for commonality is beneficial does not explicitly consider the quantitative economic effects of commonality, however it does consider several other performance metrics in great detail.

After the proposed change is evaluated it must pass through a multi-tier approval process to be implemented. The change process is an effective method for limiting the unacceptable causes for divergence. In addition, the project has taken into consideration several challenges related to the implementation of commonality, such as divergence and proprietary restrictions, and are managing these challenges through the design of the architecture and the acquisition process.

There are several similarities between the CSSS application of commonality and the seven industry case studies conducted by Boas (2008); most notably the existence of life cycle offsets and divergence. Boas found that life cycle offsets often exist and create a challenge for the development of a common system (Boas, 2008). Similarly, life cycle offsets were observed in the CSSS case study as a result of financial limitations, resource limitations, and different dates of need.

The EVA System also experienced divergence, or a reduction in commonality over time. In this case the divergence occurred for an acceptable reason, as a result of learning. The initial architecture of the system contained a Body Seal Closure (BSC) that was determined to be hazardous in the case of off-nominal entry and landing. As a result, the system diverged to eliminate the BSC risk.

There were also a few differences within the CSSS case study and previous industry case studies. The seven industry case studies conducted by Boas (2008), were predominately examples of internal applications of commonality, while NASA contracts out much of its development work. As a result, NASA must manage the contractor in order to obtain the economic benefits from commonality. This can be a challenge because the two groups often have opposing goals. NASA attempted to realign the goals of the customer and contractor through intelligent contract writing.

The EVA System's management methods seem to be working successful, but there is still much work to be done until the system is ready for launch. The project managers are expecting more divergence to occur, but it appears that there is a strong structure in place to control divergence.

4. International Standard Payload Rack

The second case study was conducted on the payload management services used on the International Space Station (ISS), with a focus on the International Standard Payload Rack (ISPR). Each of the three NASA case studies was chosen because it offered a diverse environment in which commonality had to be managed. The ISPR was chosen because, (1) it offered an example of commonality applied to a service system (the management of scientific payloads), (2) it is part of the International Space Station Program, and (3) commonality was applied internationally. The ISPR is described as part of a service system because the system is being developed in order to offer a service (to transport, house, and install scientific payloads) for a wide variety of scientific customers.

NASA holds utilization rights for the Destiny laboratory and half of both the Columbus and Kibo laboratories. Implementing commonality within the payload management system offers benefits to ground logistics, in-space logistics, and the transportation of payloads. Details of the payload management system and the development of that system are discussed in section 4.2, Payload Management System.

NASA chose to implement commonality across the laboratories in the form of standards, because it offered a formalized process to develop and control commonality. However, because there is no central authority between the international partners, all parties must agree on the design standards. More observations on the management methods used in this case are discussed in section 4.3, Observations.

4.1 Background

The initial concept to create a space station to conduct microgravity science can be traced back to the earliest days of NASA. In 1958, Congress issued a report outlining the next ten years in space and stated that the "next logical step" was to develop a space station (Harland & Catchpole, 2002). Plans to develop a space station were again placed in

NASA's path in 1969, when the Space Task Group, chaired by Vice President Spiro Agnew, urged NASA to develop the Space Shuttle for the purpose of assembling and servicing a twelve-person space station in Low-Earth Orbit (LEO) (Harland & Catchpole, 2002). However, as the estimated costs of the space shuttle increased the space station was deleted from NASA's plans. The development of the space station did not return to NASA's vision until January 25, 1984, when President Ronald Reagan announced that he was "directing NASA to develop a permanently manned space station" and that "we want our friends to help us meet these challenges and share in their benefits" (Harland & Catchpole, 2002). The proposal for the space station outlined the spending of $8 billion spread across ten years by NASA, with contributions from the European Space Agency (ESA) valued at $2 billion and Japan valued at $1 billion (Harland & Catchpole, 2002). The estimated costs amounted to very little compared to the Department of Defense's budget, which at the time was nearly $250 billion per year (Harland & Catchpole, 2002). However, even with the president's direction and help from international partners the road to developing the space station was difficult, at best.

As designs for the space station matured cost estimates doubled up to $16 billion over the same time frame, with contributions valued at $4 billion from ESA, $2 billion from Japan, and $1 billion from the Canadian Space Agency (CSA) (Harland & Catchpole, 2002). It was at this time that the first official agreements were formed between the US and each of the participating countries, in the form of Memorandums of Understanding (MOU). The MOUs created an agreement in which the international partners would each contribute one element to the space station and collectively cover 25% of operating costs over the following 25 years; ESA and Japan would each deliver laboratories and CSA would create a robotic arm. In 1988, the Senate voiced its disapproval of NASA's plans by denying the $967 million needed to begin construction of the newly dubbed Space Station Freedom, and awarded only $200 million to the project (Harland & Catchpole, 2002). Although the budget was later increased back up to $902 million by the House of Representatives, this cut marked the first of several confrontations with Congress, each causing design changes and additional delays.

With each new president, in that time, came additional requests for design changes. The largest change came in 1993, when President Clinton moved into the White House, and plans were changed to include a new partner, the Russian Federal Space Agency (RKA). The newly updated plan included three phases: phase one involved the cooperative use of the Space Shuttle and Mir in order to jump start the biomedical research program, phase two consisted of the construction of the planned NASA and Russian pressurized modules, and phase three included the addition of ESA and Japan's modules and Canada's robotic arm. There were also several design changes, including a change in the space station's inclination from 28.5 degrees to 51.6 degrees off of the equator, significantly reducing the space shuttle's cargo capacity (Harland & Catchpole, 2002). The plans called to begin assembly in 1997 and complete it by 2001, however, as Clinton's presidency continued budget cuts became all too common within NASA, causing additional delays.

It was not until January of 1998, when all participating organizations representing fifteen countries signed the Space Station Intergovernmental Agreement (IGA) that the International Space Station (ISS) was officially born. The participating organizations include the National Aeronautics and Space Administration (NASA), the European Space Agency (ESA) including eleven participating countries, the Japan Aerospace Exploration Agency (JAXA), the Canadian Space Agency (CSA), and the Russian Federal Space Agency (RKA). There had been previous agreements with Europe, Japan, and Canada, but the new ISS IGA replaced them in response to the addition of Russia as a partner and the resultant design and utilization changes. In November of that same year, Zarya, the first ISS module, was launched into orbit (Harland & Catchpole, 2002).

The ISS IGA is an international treaty for the co-operative "detailed design, development, operation, and utilization of a permanently inhabited" space station (NASA). The IGA designates NASA as the manager of the ISS, and as the manager, NASA created new MOU's with each of the participating countries. Each MOU outlines the rights and responsibilities of each participant.

The IGA and MOUs designate two segments of the International Space Station; the Russian segment and the US segment which includes the majority of the partner contributions. Figure 22 shows the expected final configuration of the ISS with the contributions of each organization indicated (as of February 2009). The Russian segment of the ISS will include (once completed) the Zvezda service module, Mini-Research Module 1 (MRM1), MRM2, and the Multi-purpose Laboratory Module. The Russian segment is completely owned and utilized by Russia. The US segment will include (once completed) the Zarya control module (a NASA contract developed and launched by Russia), the Unity node, the Destiny laboratory, the Quest joint-airlock, the Harmony node, the Columbus laboratory (ESA), the Kibo Pressurized Module and Experiment Module (JAXA), the Tranquility node, the Cupola observatory module, and a Permanent Logistics Module (Amos, 2009). The US segment is being developed and operated cooperatively by the US and its international partners ESA, JAXA, and CSA (Harland & Catchpole, 2002). The respective Russian and US segment agencies work together to coordinate logistics, including the use of Soyuz and Space Shuttle seats and the use of unmanned transfer vehicles; Russia operates the Progress transfer vehicle and Europe and Japan have developed the Automated Transfer Vehicle and H-II Transfer Vehicle, respectively. NASA has also been supporting its own Commercial Orbit Transportation Services (COTS) program to help commercial organizations provide orbit transportation services. Assembly is scheduled for completion in 2011.

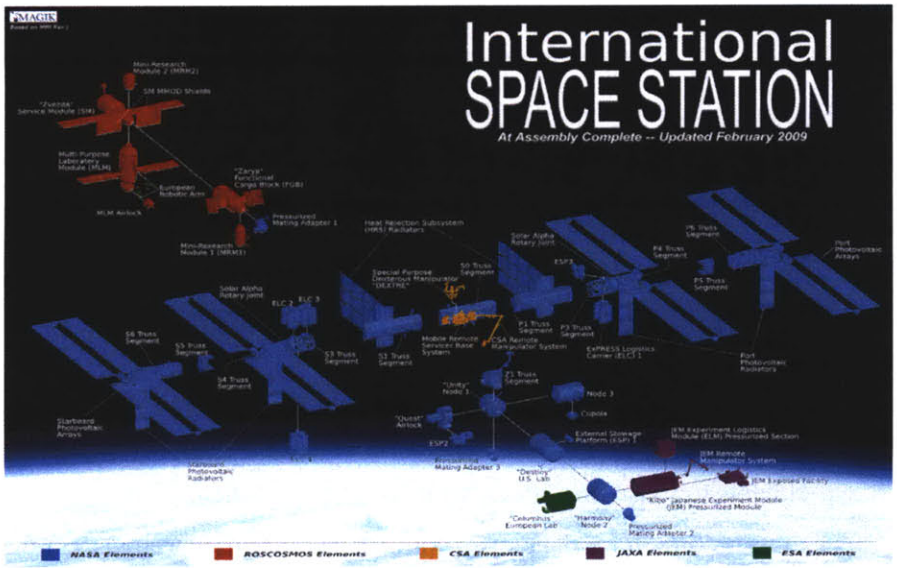


Figure 22: Final configuration of the ISS showing partner contributions (NASA)

Each of the ISS partners are responsible for their portion of the US segment; ESA is responsible for the Columbus module, Japan for the Kibo modules, Canada for the robotic arms, and the United States for Destiny, Unity, Harmony, Tranquility, Cupola, Quest, the infrastructure (including truss segments and solar panels), and transportation of the modules and infrastructure. Each partner retains jurisdiction, rights, and control over it's elements, but the utilization of each element is based on the partners overall contribution to the space station (NASA). The utilization rights for each laboratory are shown in Table 3. Russia maintains utilization rights to 100% of its modules and hardware.

Table 3: Utilization rights as outlined in the IGA and MOUs

Laboratory	**NASA**	**JAXA**	**ESA**	**CSA**
Destiny	97.7%	0%	0%	2.3%
Columbus	46.7%	0%	51%	2.3%
Kibo	46.7%	51%	0%	2.3%
Crew Time, Power, & Support Services	76%	12.8%	8.3%	2.3%

4.2 Payload Management System

One of the primary purposes for creating the International Space Station (ISS) was to create a facility in which scientific micro-gravity research could be conducted in an environment that was not possible anywhere on Earth. Today, research is still an integral part of the mission. Scientific work is currently being conducted on a range of topics including: human research and countermeasures development, physical and biological sciences, technology development, Earth observing, education outreach, and facility maintenance (NASA). In 2005, the ISS was identified as a United States National Laboratory.

The payload management system is composed of the equipment used to house and transport experiments to space, including the International Standard Payload Rack (ISPR), other sub-rack structures developed, and the vehicles used to transport the experiments. This section will discuss the original concept for the management of the payload, the evolution of the system over time, and observations on the management methods used to develop the system.

4.2.1 Original Concept and Design

The original concept for conducting science on the ISS was to fly a payload to the ISS, allow that payload to operate for its predetermined life, and subsequently return the payload back to Earth where it could either be refurbished or data from the payload could be analyzed. In this concept payloads would be continuously transferred to and from the space station as soon as space became available. During the development of the ISS, NASA was faced with the challenge of creating an efficient way of organizing, conducting, and controlling experiments. There are several factors to consider in order to operate a science program efficiently, including: (1) how to limit the amount of crew time required for installation and operations of the payloads, (2) how to efficiently transport payloads to and from the ISS, (3) how to ensure that the payloads will work efficiently in-space with no ground integrated testing, and (4) how to efficiently utilize all of the space that NASA holds utilization rights. Each of the four factors is discussed below.

Crew Time

The first factor to consider is the impact of crew time. At the beginning of the program, the ISS was completely maintained and operated by either two or three crewmembers. It is estimated that the time required to maintain the ISS is equivalent to that of 2.5 full-time crewmembers. With so few crewmembers and so much to do the majority of the crew's time is consumed by maintenance and operations, leaving little time for science. Knowing this, NASA took steps to minimize the amount of required crew time. The first step taken at the beginning of the program was to choose payloads that required relatively little crew interaction. However, the number of crewmembers on board the ISS was recently increased to six, allowing more time demanding payloads to be flown.

The second step NASA took to limit the required crew time was to develop a concept for a payload management system that would allow scientific payloads to be "plugged in" to the station similar to how appliances are plugged into the wall outlets in a home. The concept that was then developed was for a standard sized payload rack that would interface with the module through standard mechanical and consumable interfaces in designated payload rack locations. The interfaces were designed to be common enough that any rack could be plugged into any payload rack location without any customization. The concept for the standard payload rack involved the creation of a structural frame with common mechanical and consumable interfaces. Figure 23 shows the design of the payload rack structure.

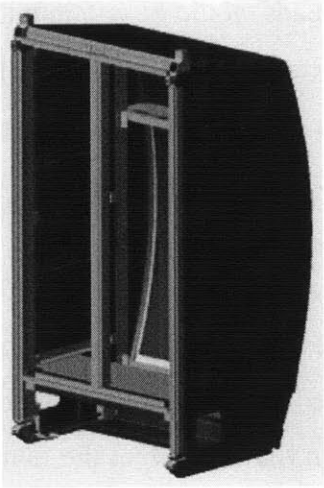


Figure 23: A drawing of an International Standard Payload Rack (ISPR)

Standard Interfaces

The Destiny Laboratory was launched in February of 2001, while payload racks will be continuously transported to and from the ISS during operations until at least 2015. As a result, scientific payloads will not be able to be tested in the laboratory before they are launched into orbit. To increase confidence in the likelihood that the payloads will operate successfully in the ISS, the consumable interfaces were defined very precisely. The more precisely defined, the more confident scientists and NASA would be in the compatibility of the systems. As long as the interface standards are met, the payload should operate successfully in space. Almost every payload rack can be placed in every payload rack location, giving the crew the flexibility to move the payloads to the most convenient location. The concept for a standard payload rack was initially used on the

Spacelab missions, however Spacelab racks had to be removed and installed on the ground.

Transportation

The third factor that was considered was the transportation of the standard payload racks to and from the ISS. The payload management system that had been designed involved racks (ISPRs) that could only be moved through the Common Berthing Mechanism (CBM), because they were too big for all of the other docking systems. As a result, the Multi-Purpose Logistics Modules (MPLMs) were developed that utilize the CBM and allow for the transportation of payload racks. MPLMs are pressurized modules that are housed in the payload bay of the Space Shuttle and are used to transport logistics, such as the standard payload racks, to the ISS. Figure 24 shows the MPLM stored in the Space Shuttle's payload bay on orbit. Once the Shuttle is docked to the ISS the robotic arm is used to remove the MPLM from the payload bay and connect it directly to the laboratory or a nearby node. The payload racks that have reached the end of their life are switched out with new payloads. Payloads that are to be transferred back to Earth are placed into the MPLM. One of the MPLMs, Donatello, was designed for powered transportation of payload racks, such as the Minus-Eighty-Laboratory Freezer for ISS (MELFI), which preserves frozen biological samples. MPLMs were the only method of transporting payload racks to and from the ISS, until 2009 when the H-II Transfer Vehicle was first launched, which also utilizes the CBM.

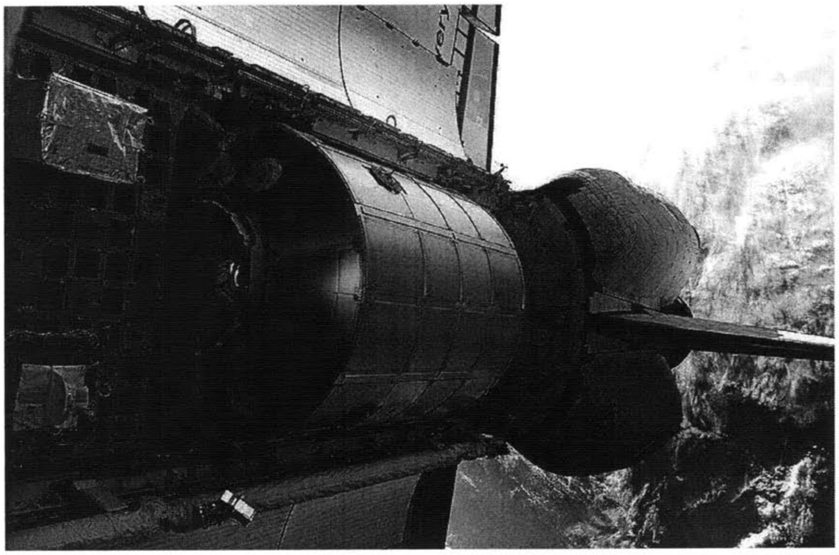


Figure 24: An image of the MPLM inside of the Space Shuttle's payload bay

International Utilization

The fourth factor that NASA considered was how to most efficiently utilize the space available for scientific research. NASA holds utilization rights for 97.7% of the Destiny laboratory and half of both the Columbus and Kibo laboratories. NASA was faced with two options: either (1) ensure that each of the laboratories could interface with the standard payload rack NASA was creating, which would enable NASA to place any of its experiment racks at any of the locations that they held the rights to, or (2) allow each partner to create their own system with different interfaces and then attempt to coordinate the different experiments and interfaces. The second alternative would be costly and create too many problems for ground and in-space logistics, so NASA began to work with the international partners to create an International Standard Payload Rack (ISPR) that could be used on all laboratories.

The Interface Control Working Group (ICWG) was created as a method of communicating and negotiating commonality issues for the International Standard Payload Rack (ISPR). The US had already created a baseline for consumable specifications, but the international partners did not want to and in some cases could not meet all of the consumable specifications domestically. Each of the international partners wanted to develop and build their own hardware within their country to stimulate the domestic economy and protect trade secrets. Some of the partners were not technologically capable of meeting the standards requested by NASA and some purely wanted different capabilities. NASA initiated negotiations with the international partners using the ICWG and worked with the international partners until they were able to come to an agreement for the initial set of baseline standards. The baseline standards are recorded in SSP 51152, the Interface Requirements Document and SSP 41002, the Interface Control Documents (NASA).

After the baseline standards were set in place, all issues were controlled through a configuration management process within the ICWG. In the configuration management system, each of the partners would write Program Release Notices (PRN) for any issues that they had with the common interface. The partners would then meet every six months to discuss all of the PRNs that had been written in the previous six months. If the requested change or waiver could financially harm any of the partners, the requester must negotiate compensation. For example, if NASA requested an additional interface for the supply of argon in the laboratories, each of the partners would review their designs and estimate the cost of adding an argon supply line; it would then be the responsibility of the requester to negotiate the terms of the change. Many of the partners did in fact request subtle changes on their modules; some of the rack locations have ports for argon, helium, carbon dioxide, and different power capabilities. In general, requests were accepted as long as the change did not affect the rest of the interface. Many of the modules have unique interfaces, but any ISPR can still be installed at any ISPR location.

After the specifications were set, the Destiny laboratory designs were finalized and the laboratory was flown in 2001. The Destiny laboratory can hold 24 total racks, 13 of which are ISPRs, while 11 are system racks. System racks are racks that control the ISS systems, including: habitation; the Crew Health Care System (CHeCS); Extravehicular Activity (EVA); the Environmental Control and Life Support System (ECLSS); computers and data management; propulsion; Guidance, Navigation, and Control (GNC); communications; the Thermal Control System (TCS); and the Electrical Power System (EPS). Boeing built the original ISPRs using composites, however JAXA developed and manufactured their own ISPRs domestically using aluminum, and eventually sold ISPRs to ESA.

4.2.2 Evolution of the Design

The original concept for science on the ISS was to continuously transfer and switch out entire ISPRs using the Space Shuttle and MPLM. In the original concept, each rack was flown on the Space Shuttle, operated in the ISS, and returned to the ground when operations were complete. However, this concept was revisited in 2003 after the Columbia accident. The Columbia accident created serious concerns about shuttle safety, and eventually led the Columbia Accident Investigation Board (CAIB) to suggest that the Shuttle be retired in 2010 or undergoes a thorough recertification process. As a result, the Shuttle is slated for retirement in 2010 and the ISS has had to evolve to fit with the schedule changes. First, NASA was forced to cut the Centrifuge Accommodations Module and the Habitation Module in order to relieve the Shuttle launch schedule and budget. The concept for the transportation of ISPRs completely relied on the frequent flight of the Shuttle because entire ISPRs could only be transported using the Shuttle and MPLM. Other vehicles utilize the Low-Impact Docking System (LIDS) and the Androgynous Peripheral Attach System (APAS) hatches, which are not large enough to accommodate the ISPRs. So, NASA needed to find a more flexible way of transporting science experiments to the ISS in order to maintain a successful science program past 2010. The solution laid within the EXpedite the PRocessing of Experiments to Space Station or EXPRESS racks, shown in Figure 25.

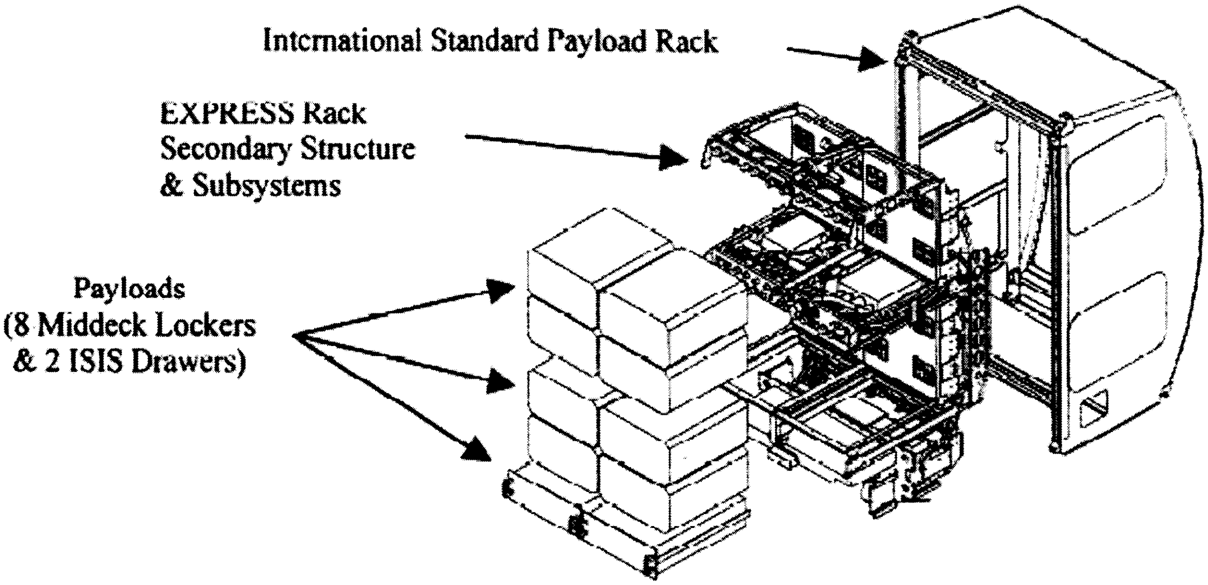


Figure 25: Image of the EXPRESS rack, displaying the available volume(Sledd, 2000)

The EXPRESS rack uses an ISPR to connect to the module, but it also sub-organizes the ISPR into a smaller arrangement of payloads. The EXPRESS rack is sub-divided into eight Space Shuttle middeck locker sized payload spaces and two International Subrack Interface Standard (ISIS) drawer payload spaces, as opposed to entire ISPR sized payloads. The available space can be subdivided in any way necessary, and all of the ISPR consumables can be used, with the only change being a drop in electrical voltage from 120 V to 28 V. In all ISPRs the consumables are connected at the back of the rack, but the EXPRESS rack reroutes the consumables to the front so that the crew can easily interchange experiments and reconnect the necessary consumables without removing the rack. The EXPRESS rack is designed to stay in orbit while experiments are interchanged within the rack.

The EXPRESS rack was originally designed because there was a greater demand for smaller experiment packages than the over 40 cubic feet of an entire ISPR (Hashimoto, Fukatsu, Ano, Mizuno, Hashimoto, & Tokumura, 1998). The EXPRESS rack design retains the benefits of the ISPR by offering ground and in-space logistics benefits, but also offers additional flexibility for the transportation and size of the experiments. Eight EXPRESS racks were built and six are already in operation on the ISS. The EXPRESS rack allows payloads to be transported to the ISS in an MPLM, the Shuttle middeck, a

Russian Soyuz, the Russian Progress, ESA's Automated Transfer Vehicle (ATV), or Japan's H-II Transfer Vehicle (HTV). The ability to transport payloads in the Shuttle middeck also enables powered transportation of payloads for the first time, because the Donatello MPLM was never flown. The General Laboratory Active Cryogenic ISS Experiment Refrigerator (GLACIER) is a -160 °C freezer designed for operation in an EXPRESS rack, but can also be used to transport frozen samples while powered in the shuttle middeck. The COTS transportation vehicles in development are also being designed to transport shuttle middeck locker and ISIS drawer sized payloads and support the powered transportation of payloads such as the GLACIER system.

Several other racks were also created using the ISPR that are common to the EXPRESS rack, including the Human Research Facility (HRF) racks and the Habitat Holding Racks (HHR). Both of these racks are created using ISPRs, designed to accommodate ISIS drawers, and use the same basic avionics as the EXPRESS racks.

4.2.3 Results

The International Space Station has been continuously manned and operated since November 2nd, 2000 and it will continue to be operated until at least 2015. Construction of the ISS is scheduled for completion in 2011 as the Space Shuttle retires. After the shuttle retires, the ISS will continue to be operated with crew transport coming from Soyuz and the Orion vehicle, while the Progress, ATV, HTV, and COTS transfer vehicles provide logistics.

The ISS has continued to progress, with an increase in the number of crew from the initial two or three up to the current six. There are currently 33 ISPR locations dedicated to science, thirteen on Destiny, ten on Columbus, and ten on Kibo; the US has utilization rights to 23 of the ISPR locations. New ISPRs can be flown as needed on either HTV or an MPLM, while sub-rack payloads can be flown on any of the vehicles. As of June 1st, 2007, 105 scientific studies had been completed or were nearing completion (Uri, 2007). This number has continued to rise as more crew time has been dedicated to support science and as more science payloads are brought up to the station.

Flexibility in the transportation of payloads has proven to be extremely important to NASA's science program as was seen when the Shuttle was grounded following the Columbia Accident. Payload flexibility is also important because NASA's mission can also change unexpectedly. This was shown when the Vision for Space Exploration (VSE) was announced in 2004 and NASA was obliged to refocus its ISS utilization plans. NASA shifted its focus to include three broad areas: (1) astronaut health and countermeasures for long-term space exploration, (2) research and developments for future exploration mission needs, and (3) operation practices and procedures for long-duration missions (Robinson, Thumm, & Thomas, 2007). As a result of the flexible design, NASA's payload systems, which include the ISPRs and EXPRESS racks, have been able to efficiently accommodate the shift in utilization and transportation concepts.

4.3 Observations

The following section outlines the observations made in the course of the case study related to the management of commonality, including the challenges related to the management of commonality and the methods used to manage the application of commonality. The management observations are organized based on the three critical tasks for managing commonality: the management of the *identification* of opportunities, the *evaluation* of opportunities, and finally the *implementation* of opportunities that are deemed beneficial.

4.3.1 Challenges

Commonality will not occur in a system by chance, but must be actively sought out and managed in order for it to be applied to a system. There are several challenges related to the management of commonality, the challenges observed in the ISPR case study are outlined below.

Life Cycle Offsets

Offsets create several problems for management related to the application of commonality, discussed in section 2.2. The Destiny Laboratory was launched in 2001, with the Columbus and Kibo laboratories following in 2008. The laboratories are faced with the challenge of developing common interfaces and standards despite this offset. In this case the laboratories were offset because of different dates of need and constrained to the shuttle's launch schedule, which drastically changed after the Columbia accident.

Divergence

Divergence is the decrease in intended commonality between systems during development. Even though commonality was implemented primarily in the form of standards, divergence did occur in the form of wavers.

Lack of Central Authority

One of the unique challenges for the creation of the common payload management system was to implement commonality between international partners in which no central authority exists to mandate or control designs. This challenge was not observed in any of the industry case studies of commonality management.

4.3.2 Management Methods

This section discusses the observed management methods from the ISPR case study. The section is organized to explain the management methods used as they relate to the three tasks for applying commonality successfully: managing the *identification* of opportunities, *evaluating* opportunities, and *implementing* those opportunities that are deemed beneficial.

Manage the Identification of Opportunities

The first task involved in managing commonality is to identify opportunities for commonality. Several factors were observed to be important for the identification of commonality, including the timing in which commonality was identified and the development offset of the common systems.

Timing

Observation: The opportunity to create a common payload management system was identified early enough that NASA was able to negotiate designs and interfaces with the international partners.

In the previous NASA case studies, the timing in which an opportunity was identified and subsequently evaluated and implemented was found to be a critical factor. Timing was also seen to be a key component for success in the ISPR case study, because the partners must negotiate towards the creation of a common design or standard. NASA was able to initiate communication and negotiate designs with the international partners in large part because they began communication early, before the development of the Destiny laboratory was complete or mature enough to prohibit changes. It would have been extremely costly to make changes after designs had matured and cost prohibitive once hardware was manufactured or flown.

Evaluate Opportunities

Commonality does not always benefit a system, but instead often implies a trade. It is in this fact that the second task in managing commonality exists, evaluating opportunities for commonality. The benefits and penalties to the entire system's life cycle should be evaluated to determine whether or not the opportunity is beneficial and should be implemented. However, opportunities for commonality often affect the system in unexpected ways, which can make the evaluation difficult. For most companies the benefits and penalties can be represented monetarily, however for an organization such as NASA in which the driver for commonality is not monetary profit, it is equally important to evaluate the non-monetary benefits.

Non-monetary Benefits and Costs

Observation: Non-monetary benefits to the creation of an International Standard Payload Rack were a significant driver in the development of the system.

The creation of a common system in industry is always driven by monetary benefits or profits, either to decrease costs in order to increase profit or to reach new markets to increase revenue and as a result profit. However, in the case of a government organization profit is not a driver, instead the driver is to maximize the benefits for a given cost; both the benefits and costs can, and often are, non-monetary. In the case of the ISPR the goal was to maximize the amount of science that could be conducted safely on the ISS. Costs to the system include the costs of ground logistics and transportation, but also in-space logistics. On the ISS crew time is an extremely scarce resource because it takes an estimated 2.5 crew working full-time to maintain and operate the ISS, leaving little time for science. As a result, one of the primary concerns when evaluating the opportunity for a common payload rack were the effects on in-space logistics and crew time. The ISPR payload management system was finally implemented because it offered a simpler system for the ground logistics, in-space logistics, and the transportation of scientific payloads using a combination of ISPRs, MPLMs, and other payload racks like the EXPRESS rack.

Despite the extensive evaluation of non-monetary benefits, little to no quantitative economic analysis was observed to determine what the quantitative economic benefits and penalties were to the system. This problem has been observed in all three NASA case studies and seven industry case studies conducted (Boas, 2008). As a result, Chapter 7 presents the work completed to date to better evaluate opportunities for commonality.

Implement Beneficial Opportunities

The last task for the management of commonality is the successful implementation of opportunities that are deemed to be beneficial. Once the decision has been made to implement commonality there were several methods observed to maximize the benefits. The methods observed in the course of the ISPR case study, include the importance of negotiation and compromise, the use of standards, and flexibility in the system.

Accelerating Development

Observation: The international partners were able to begin communication early and to some degree develop in parallel to negotiate commonality.

The Destiny laboratory was the first laboratory developed for the ISS, launched in 2001, with Columbus and Kibo following in 2008. In industry case studies, it was observed that when the first system is designed, the future systems are often unknown and unrepresented in design decisions (Boas, 2008). In this case, however NASA and the international partners were able to work together and negotiate to create a common system despite the large life cycle offsets. In order to enable intelligent negotiations of the interface the international partners accelerated portions of their development.

In the industry case studies conducted by (Boas, 2008), it was found that in most if not all cases common systems had life cycle offsets. There are several factors that contribute to the existence of offsets, however financial constraints, one of the most significant factors, do not apply to development by peer organizations. Therefore, it is conceivable that portions of development could in fact be conducted in parallel, greatly simplifying communication and integration between the separate development groups.

Standardization

Observation: NASA and the international partners implemented commonality primarily in the form of standards.

Standardization is the creation of some degree of similarity or commonality by an authority or general consent in order to obtain benefits. Implementing commonality in the form of a standard is especially useful for the case in which no central authority exists because it allows general consent to be formalized between developers to create a common design, interface, or process. Once the standard is in place and agreed upon, any divergence from that standard will also require general consent or an agreement among the partners.

NASA and the other international partners have created several standards, including standard interfaces, standard design specifications, and standard operations. The standards have proven extremely helpful in regulating and negotiating design issues among the groups. However, standards are often limited to lower levels of design.

Compromise

Observation: Compromise was an essential element when creating commonality between the international partners.

When commonality is being applied between peer organizations in which no central authority is in place, compromise is essential. In the ISPR case study, all of the participating organizations are able to obtain benefits from commonality. NASA will be able to place their racks at any location that it has rights to on any module, while JAXA and ESA will be able to transport their payloads on NASA's vehicle and reuse NASA's development work. However, the international partners could not meet all of the standards that NASA proposed, so the groups had to be willing to compromise in order to create a successful common system. However, in some cases negotiations were driven by political factors just as much as it was driven by engineering factors.

Flexibility

Observation: NASA's original concept for payload management relied heavily on the Space Shuttle to transport ISPRs. The EXPRESS racks were later developed to offer flexibility in the transportation and size of experiments.

The original concept for science operations, involved the transportation of ISPRs to and from the ISS using MPLMs and the Space Shuttle. Once on orbit, the ISPRs would be operated for their desired lifespan and then swapped out with a new ISPR. From 2001, when the Destiny laboratory was launched, until 2009, the only way to transport ISPRs was with a MPLM because it was the only vehicle that used the much larger CBM to dock with the ISS. NASA and its international partners found themselves highly reliant on the space shuttle in order to conduct science on the ISS. This fact, along with a growing demand for smaller experiments, led NASA to create sub rack structures, such as the EXPRESS rack. The EXPRESS rack is a sub-organized ISPR that allows shuttle middeck locker sized experiments to be transported to and from the ISS, as opposed to entire ISPR sized payloads. The EXPRESS rack offers greater transportation flexibility as well as experiment flexibility because experiments of all sizes can be transported on a number of vehicles.

4.4 Conclusion

Commonality has the potential to tremendously benefit a system if it is applied efficiently. A case study of the International Standard Payload Rack (ISPR) was conducted in order to determine how commonality is currently being managed within NASA and to identify the best practices and challenges associated with commonality.

The creation of a standard payload rack was initiated because NASA was attempting to maximize the efficiency in which science is conducted on the ISS by: maximizing the utilization of available space, reducing ground and in-space logistics, and simplifying the transportation of payloads. These factors eventually led NASA to negotiate with the international partners towards the creation of an International Standard Payload Rack (ISPR) because NASA shares experiment space on both the Kibo and Columbus modules.

The international partners were able to work together early in the design process to come to a formal consensus and standardize the consumables interfaces, rack designs, and several aspects of the module design. The common rack and consumable interfaces allows NASA to place ISPRs on any of the laboratories, including five ISPR locations on each the Columbus and Kibo laboratories and thirteen ISPR locations on Destiny. Additionally, NASA created eight EXPRESS racks to offer greater flexibility in regards to transportation method and experiment size. The EXPRESS racks offer interfaces that are common with the Shuttle middeck lockers and will also be common with the future COTS transportation vehicle.

The seven industry case studies previously conducted by Boas (2008) found that in nearly all cases life cycle offsets existed. Life cycle offsets also exist in the ISPR case, however the international partners were able to conduct portions of the development in parallel because the offsets were not created by financial constraints. The standards were created by a general consensus among users, which required each of the partners that were agreeing on the standard to mature the design sufficiently enough to understand the impacts of the standards on their system. Divergence was also observed in the ISPR case study in the form of waivers.

The ISS has been continuously manned since 2000, and in that time the crewmembers have been able to conduct an incredible amount of science. Additionally, scientific research has recently ramped up because the Columbus and Kibo laboratories were launched in 2008 and the number of crewmembers was increased to six in 2009. The efficient execution of scientific research on the ISS thus far can be attributed to the international collaboration and the well-conceived operations concept for science payloads, including the use of ISPRs, EXPRESS racks, and associated transfer vehicles.

5. Core Flight Software System

The third of the three case studies was conducted on the Core Flight-Software System (CFS). Each of the case studies was chosen because it offered a diverse environment in which commonality had to be managed. The CFS was chosen because (1) it offers an example of commonality applied within a software system, (2) it is part of the unmanned space flight program, and (3) because development for the CFS was conducted in-house.

Goddard Space Flight Center (GSFC) creates many of NASA's unmanned spacecraft, but in recent years there has been growing competition for the development of these unmanned spacecraft. As a result, managers are seeking new methods to decrease costs, leading the Flight Software Branch to identify common flight software as an opportunity to decrease life cycle costs. The resultant system is called the Core Flight-Software System.

The flight software branch used a variety of methods to develop the Core Flight-Software System. Design and architecture decisions for the CFS were educated by a heritage analysis, in which recent satellite architectures were analyzed to determine whether or not a particular function should be developed to be common. The system was then developed in a parallel development stream, allowing the engineers to develop the system without compromising system goals to meet a particular satellite's schedule constraints. The system was then deployed in stages, to obtain benefits sooner. Specific details of the management methods observed in the course of this case study are discussed in section 5.3, Observations.

5.1 Background

The following section presents the relevant background to the Core Flight-Software System (CFS) case study. The background includes information on the development of common software systems, the Goddard Space Flight Center, and flight software systems.

5.1.1 Common Software

Computers and the programs that run them (i.e. software) are becoming a fundamental part of any engineering system and the development, production, and operations for that system. As a result, software is also becoming a larger portion of the total life cycle cost. This trend can be visualized when looking at the growth of the software industry as a whole; in 2002 the global revenue of the largest 500 software companies was $288 billion, by 2007 that number had risen to $451.8 billion (Desmond, 2008). This trend also holds true in the aerospace industry, the space shuttle software contains about 400,000 lines of code and NASA spends approximately $100,000,000 per year to maintain the software (Leveson N.).

Computer software is also an integral part of a satellite system. Each satellite is run by a flight-software (FSW) system that can be designed to control communications, attitude and orbit control, navigation, fault detection, mission management, and payload management. FSW systems offer more capability now than ever before, but they also have the potential to drive up life cycle costs. Commonality has been identified as a valuable method of controlling these life cycle costs.

Commonality has already been used successfully in the computer and software industry for a few decades to dramatically cut costs. In the early years of the computer industry, the 1970s and 1980s, software applications were designed to operate on a specific set of hardware, sometimes referred to as the software's platform (Meyer & Seliger, 1998). In order to run the same software on a different platform, the software had to be adapted for the new platform. This method of development dramatically changed with the introduction of the Operating System (OS) in the late 1980s (Meyer & Seliger, 1998). The OS introduced a common interface layer between the computer and software applications that eliminated the need to customize software applications for each computer; as long as the OS could be run successfully on the hardware, the software application could also be run successfully on the OS and underlying hardware.

In more recent years, this same method of creating common software interfaces has been used to not only reduce costs, but to also make the software customizable and flexible for specific niche markets. The Flight Software Branch at Goddard Space Flight Center (GSFC) is attempting to use a similar architecture concept to create a common Flight Software (FSW) system that is flexible enough to be used on all future satellites.

5.1.2 Goddard Space Flight Center (GSFC)

Goddard Space Flight Center (GSFC) was originally established on May 1, 1959 as NASA's first space flight center. The center was originally involved in manned programs such as Project Mercury, but after the establishment of the Johnson Space Center (JSC) in 1963, GSFC became primarily focused on unmanned spacecraft. The center currently employs a combination of scientists and engineers to develop unmanned scientific spacecraft that predominately operate in Earth's orbit. Some notable missions include the Hubble Space Telescope, Interferometric Evaluation of Glacier Rheology and Alterations (INTEGRAL), the Earth Observing System (EOS), the Solar and Heliospheric Observatory (SOHO), the Small Explorer (SMEX) satellites, and most recently the Lunar Reconnaissance Orbiter (LRO).

In the past, it was common practice to optimize each satellite by developing unique hardware and FSW to meet the particular requirements of each satellite. In addition, there was little common management in place between the several different FSW groups that existed across GSFC; each of the groups developed FSW systems independently. It was not until the early 1990's with the introduction of the SMEX satellites that the development practices were re-evaluated.

The SMEX satellites were intended to be inexpensive satellites, similar in size and capability to each other. At the time that the SMEX missions were planned it was conceived that a common FSW system could be used for all SMEX missions, thus reducing the cost and time to develop each satellite. As a result, the FSW system for SAMPEX, one of the first SMEX missions, was designed with the intent that it be reused. The FSW that was developed for SAMPEX was the first software developed at GSFC to run on a commercial processor (i386) and a commercial operating system (VRTX), both major steps for the program. After SAMPEX was launched however, the separate FSW groups across the center reactively reused the common system altering it to their satellite's specific needs as demonstrated by the evolution of the system, shown in Figure 26. This reactive reuse caused the system to become increasingly customized, diverging into several very different software systems. As divergence occurred and less commonality existed between the satellite systems the benefits also continued to decrease.

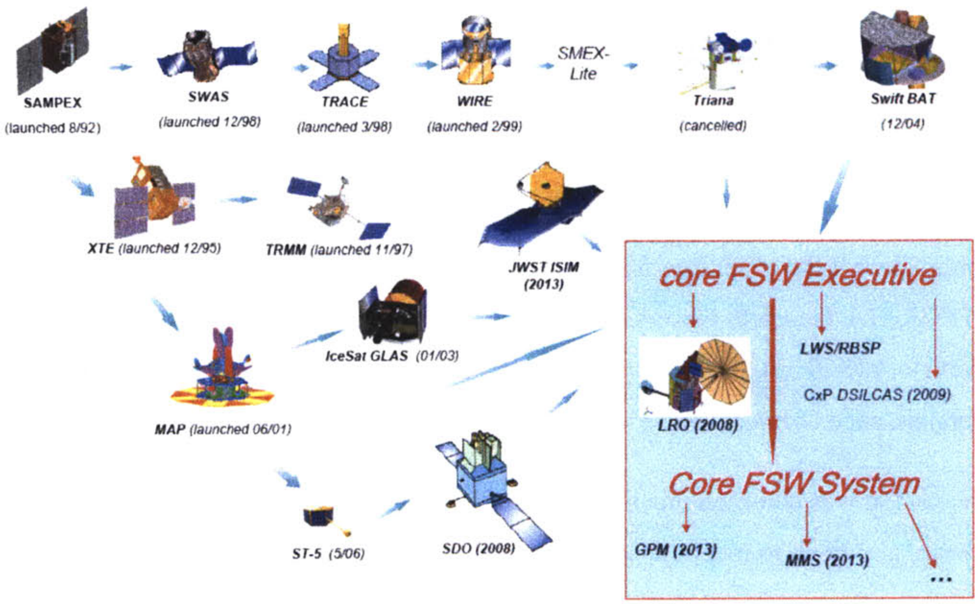


Figure 26: Heritage analysis revealed the evolution of flight software system architectures over time (Wilmot, Core Flight Software System (CFS) Presentation)

5.2 Core Flight-Software System

The following section discusses the Core Flight-Software System (CFS), including the requirements and development of the flight software system, the evolution of the system, and the current architecture of the system.

5.2.1 System Development and Requirements

Although the attempt at commonality developed for the SAMPEX satellite quickly diverged, the idea to create a single Flight-Software (FSW) system that could become common between satellites stuck. The Core FSW System (CFS) is the second attempt to create a common FSW system. Development for the CFS was initiated by two main factors: the reorganization of GSFC and increasing competition for the development of satellite systems.

When GSFC was recently reorganized, all of the flight software engineers from across the center were moved into a single branch, the Flight Software Branch (Code 582). The reorganization put all of the flight software engineers under the same management and opened up lines of communication that did not exist when the SAMPEX FSW system was developed.

The new branch management was faced with increasing competition because additional centers and laboratories had begun to compete for the development of unmanned satellites, including: the Jet Propulsion Laboratory, the Applied Physics Laboratory, Ames Research Center, and Langley Research Center. As a result of the increased competition, the branch management began looking for ways to cut costs and stay competitive. The branch then identified the opportunity to utilize a common FSW system to dramatically cut costs. A study was initiated to determine which of the several FSW architectures that had evolved since SAMPEX should be used as the common system. It was determined that each of the architectures was advantageous in particular ways, but none of the architectures could be reused effectively on a diverse set of satellites. Consequently, the branch decided to create a new system that could be effectively reused on future satellites, in hopes that the new system would give the branch a competitive advantage over other flight software developers.

The first step in developing the common system was to identify the requirements. One of the challenges related to the creation an intended common system is that the requirements of the future satellites are unknown. To address this challenge the Flight Software Branch conducted a heritage analysis to analyze past satellite FSW systems and educate architecture decisions for the CFS. During the heritage analysis each individual function or element was presented to a group of stakeholders, then through deliberation each function was organized into what should be common and what should be unique. The group of stakeholders included product leads, technical experts, test engineers, and maintenance engineers. The heritage analysis included satellites created by GSFC over the past 15+ years, and spanned 14 missions. Figure 26 shows many of the satellites that were evaluated as part of the heritage analysis. The engineers attempted to develop a system that was flexible enough that it could have been used on the past satellite systems, so that it was more likely to work successfully on future satellites. After several iterations, the process resulted in a set of requirements and a system architecture that could be common for future satellites.

Before the CFS system could be developed there were three challenges that the management would need to overcome in order to implement the common system: (1) how to fund the development and maintenance of the system, (2) how to meet current schedule and cost requirements without compromising commonality, and (3) how to maximize the benefits in future use. The management methods for each of these three challenges are described below.

When life cycle offsets exist, commonality requires an investment during the development of the first satellite FSW that no individual satellite was willing to pay, because the benefits would fall on future satellites. As a result, the initial effort to develop the common system was made separate of a particular satellite application, with funding coming from some branch resources and in part by individual engineers working on their own time to begin the heritage analysis. However, as the development of the system gained speed the branch diverted additional funding towards the development and eventually, GSFC and NASA headquarters also contributed funding. Now that the system is in operation and being applied to additional satellites, the system has become self-sustaining. Funding required for system maintenance and further development is generated through a “tax” paid by users, on the order of one or two full-time equivalents. The system is currently being used on multiple missions at GSFC and will likely be used by the Applied Physics Lab (APL), Ames Research Center (ARC), and Langley Research Center (LRC).

The next challenge was to develop the common system while still meeting the schedule, cost, and performance goals of the current satellite application. Developing a common system takes longer than an independent system would, because the common system is more complex than independently developed systems. The flight software engineers' previous experience, developing the SAMPEX system, proved that development of the CFS would be compromised if developers ran into a schedule deadline for the current satellite application. The solution was to develop the CFS independent of a specific satellite application. This approach was used to avoid impending divergence caused by an individual satellite's needs. As a result, portions of the CFS were deployed and implemented on a satellite as they became mature. However, an independent development flow can only be used if the developers also have independent funding to support an independent development effort.

The last challenge that management faced was to maximize the commonality benefits. One common misconception with software commonality is that if code is reused than the system will benefit. Developing a flight software system involves several tasks that can be broadly lumped into four categories: the development, coding, testing and integration, and maintenance. In general coding is the least expensive task; and in many cases making changes to existing code is more time consuming than recoding a system from the requirements (Leveson & Weiss, 2004). A goal then was made to not only reuse the code, but to reuse as much of the development process as possible. This goal was addressed by creating the system in the form of a library that includes the code for the cFE, OSAL, messaging middleware, CFS applications, requirements documents, test documents, and a test suite that can be used with the CFS.

5.2.2 CFS Architecture

Now that the design rationale and process for creating the CFS has been explained, the architecture of the CFS will be explained in the following section. During the deliberation process three critical system goals were identified and implemented that significantly shaped the system architecture. The system goals were interoperability, scalability, and operational flexibility.

The first goal, was to make the system interoperable with various hardware platforms and operating systems. This goal was accomplished by developing a layered architecture with common interfaces between each of the layers and the creation of the Operating System Abstraction Layer (OSAL). Figure 27 shows a diagram of the CFS with each of the layers, including the Operating System Abstraction Layer (OSAL), the core Flight Executive (cFE), a messaging middleware, and a portfolio of applications (labeled "SW Components") that can be used to customize each FSW system.

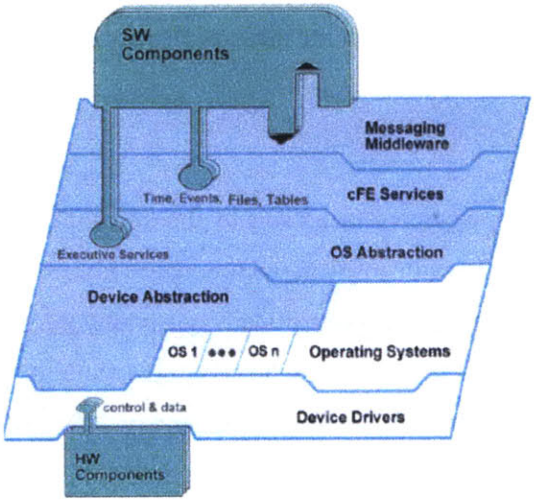


Figure 27: CFS Architecture layering, the blue layers indicate the common core, while the green indicate the customizable sections (although there are several common SW applications). Divergence has occurred, causing the device abstraction layer to no longer be a part of the core. (Wilmot, 2008)

Creating the system as a layered architecture allows each layer to work together as long as they have the common interface. For example, the cFE offers the minimal services for each satellite, and runs on the OSAL. The OSAL is a common operating system interface that can in turn be run on a wide variety of operating systems. The OSAL is currently designed to support VxWorks, RTEMS, Linux, Cygwin, and Mac OS X. As long as the chosen OS is one of the supported OS's, the OSAL and in turn the cFE will run on the satellite. Additionally, the CFS applications and the messaging middleware are able to run on the cFE system. The interoperability has the additional benefit of allowing developers to run the cFE and CFS applications on almost any platform, including a PC, allowing development and testing to be done in a simplified manner. The OSAL is open source software that is available to the public.

The second goal, was to make the CFS scalable to satellites of all sizes. To accomplish the scalability goal, the cFE was developed to offer only the minimal services that each satellite would need, and the ability to add CFS applications. The small footprint of the cFE allows the system to be used on small satellites, but the creation of CFS applications offers the ability to scale the CFS to larger more complex satellites (Wilmot, 2008). In order to minimize the footprint and memory utilization the cFE only offers the following services:

- Software bus services - a publish and subscribe messaging middleware for applications
- Time services - provides time for all other systems and applications
- Executive services - provides the ability to start, stop, suspend, and resume applications
- Event services - handles all ground status messages
- File service - provides read and write access for file headers
- Table services - provides a table of data values that can be loaded or dumped at the same time

Any additional functions or services that are required by satellites are made into common CFS applications. Each of the applications has standard application programmer interfaces (API) and can be added and removed from the running system (Wilmot, 2008). These applications are created to be as independent from other applications as possible so that each one can be used and updated with little interference to other applications. Figure 28 is a diagram of the CFS messaging middleware and CFS applications that shows the relative independence of each application and scalability of the system. The CFS applications include but are not limited to: limit checker, memory dwell, data storage, memory manager, checksum, telemetry monitor, and many others. Additional applications can be added to the system as long as they adhere to the interface standards.

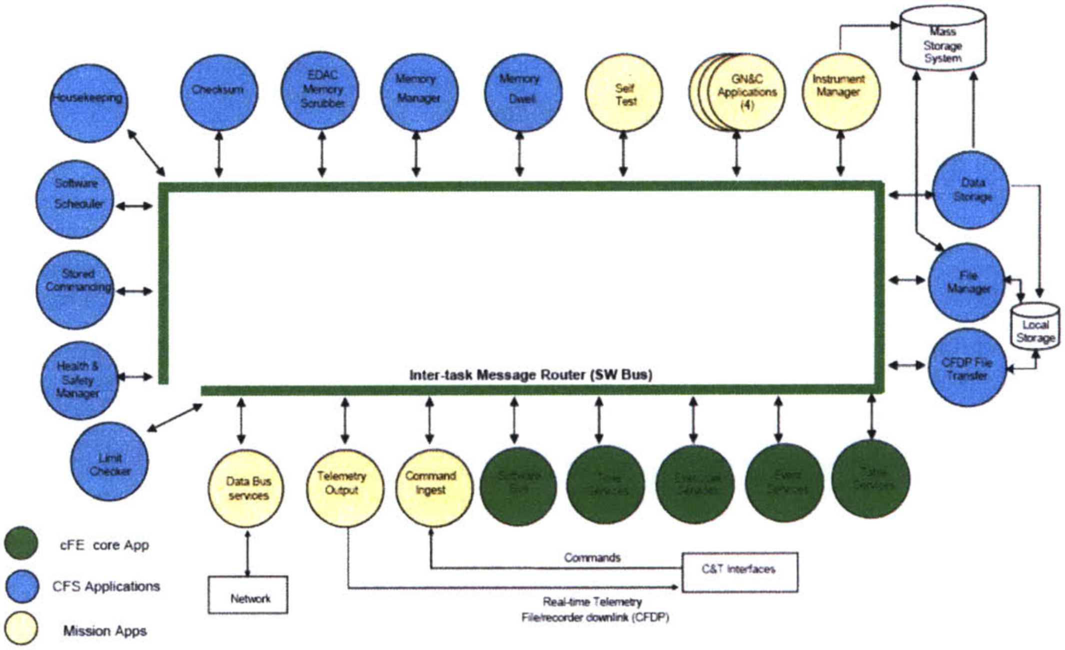


Figure 28: A diagram showing the CFS applications and the interactions with the messaging middleware (Wilmot, Core Flight Software System (CFS) Presentation)

The final goal was to develop the system to be operationally flexible, and allow for flexible maintenance and updates. The heritage architectures were designed with a compile-time environment in which all interactions had to be defined at compile time, before launch. Any updates to the system would require compiling, linking, and reloading the entire system, a lengthy and undesirable process (Wilmot, 2008). With a complex system like the CFS, it was foreseeable that the system may need maintenance. In order to avoid the undesirable compile-time environment the system was designed to work in a run-time environment, in which updates can be applied without compiling, linking, and reloading the entire system. The run-time environment is enabled by the modularity of the applications, and the ability to stop and run each of the functions that are not in use at the time (Wilmot, 2008). Therefore, a portion of the software can be stopped, altered and rerun without having to shutdown the entire system. Having the system designed in this way increases flexibility and lowers maintenance costs.

5.2.3 Results

As stated previously, the system was developed independent of a specific satellite and thus deployed in stages as each portion was completed. The first mission to use any portion of the system was CHIPSat, which was launched with a cFE prototype. The first mission to use the OSAL was the Solar Dynamics Observatory (SDO). SDO did not use other parts of the CFS system because the other portions were not mature at the time of SDO's development. Next, LRO became the first mission to utilize the modern cFE component of the system and the OSAL together. At this time it is unclear if LRO realized any economic benefits from using the cFE system because the team had to invest time and money to mature and test the cFE, however, it does not appear that the project was negatively impacted either. The next two Goddard missions, GPM and MMS, are both baselined to utilize the full CFS. Both project teams are being taxed to further mature the CFS system, however, they are also seeing the immediate benefits. For example, the basic software system, which includes the cFE, OSAL, and several applications, was prototyped and undergoing preliminary testing within a week where as in the past it could have taken several months to create a working prototype from scratch. The quick start-up has allowed them to create and test very rapidly.

All components in the library were developed in-house by NASA and are thus property of the government. As a result, the library must be made available to all other NASA employees and government organizations. The only restrictions for use of the system is that the user must sign an agreement ensuring that they will not share the system with others and that they will provide feedback to the developers, allowing the system to be maintained and improved. As a result, the CFS now has several external users, such as Ames Research Center, Applied Physics Lab, and Langley Research Center. Originally, the requirement to share the library with others was seen to be detrimental because it would allow GSFC to lose their competitive edge, however thus far users seem willing to pay the tax to further develop and maintain the system, resulting in a more mature system for all users.

5.3 Observations

The following section summarizes the observations made in the course of the case study, which include challenges and management methods for applying commonality. The management methods are organized into the three critical tasks for managing the application of commonality: the *identification* of opportunities for commonality, the *evaluation* of opportunities, and finally the *implementation* of opportunities that are deemed beneficial.

5.3.1 Challenges

Commonality will not occur in a system by chance, but must be actively sought out and managed in order for it to be applied to a system. There are several challenges related to the management of commonality, the challenges observed in the CFS case study are outlined below.

Lack of Coordination

After the development of the SAMPAX flight software system, each of the separate flight software groups spread across the center reactively reused the system until several very different architectures emerged. It was observed by many of the developers that there was little to no integration or communication between each of the different groups to coordinate the evolution of the system or the manner in which the system was reused. Another attempt at creating a common flight software system was not made until the center was reorganized and a central management authority was in place to drive the development and evolution of the system

Life Cycle Offsets

Goddard Space Flight Center is in constant development of several different satellites, each of which are offset from each other in time. Offsets between each of the systems cause an upfront penalty for the development of a common system while future systems acquire the benefits. No individual satellite was willing to pay the necessary investment to create a common system because all of the benefits would fall on future users.

Offsets also cause the future systems to be uncertain. One problem that was observed in the past is that development and design decisions are heavily weighted towards the requirements of the current system because the future systems are uncertain.

Divergence

During the development of the SAMPEX mission, it was observed that schedule and budget constraints forced the system to take the "path of least resistance" and diverge. Development requires an upfront investment of time and money, and if that investment isn't made then divergence will occur and less commonality will be developed.

5.3.2 Management Methods

This section discusses the observed management methods from the CFS case study. The section is organized to explain the management methods used as they relate to the three tasks for applying commonality successfully: managing the *identification* of opportunities, *evaluating* opportunities, and *implementing* those opportunities that are deemed beneficial.

Manage the Identification of Opportunities

The first task involved in applying commonality is to identify opportunities for commonality. Several methods were observed to manage this task and are discussed below as either active identification methods or passive identification methods. Active identification methods are the methods used by management for the specific purpose of identifying commonality, while passive identification methods enable engineers to identify opportunities for commonality.

Timing

Observation: Timing in which commonality was identified was less important in this case for the successful development of commonality because the system was developed in an independent development stream.

In the previous NASA case studies, the time at which an opportunity for commonality was identified, evaluated, and finally implemented was found to be a critical factor in determining the degree of success possible, because as time progresses the ability to affect the design and life cycle costs decrease. However, the CFS system was developed independent of a particular application in order to maximize the potential benefits and long-term success. The creation of the independent development stream causes the timing to be less critical in determining the benefits from commonality, but timing is just as critical for the application of commonality to an upcoming system.

Active Identification

Active identification methods are the management steps taken for the purpose of identifying opportunities for commonality.

Management Support

Observation: Managerial and financial support from the branch, center, and headquarters allowed the CFS system to be developed independent of an application.

Development of the CFS system was initiated and enabled by the support of management. This support started once GSFC was reorganized, moving all flight software engineers under the same branch management. The branch management then initiated a study to determine which FSW architecture could be used on all future satellites. Branch managers then supported the process by encouraging engineers to participate in the development and by contributing financial support for development. However, even with the managerial support development progressed slowly because of insufficient funding. It was not until the development received additional funding from the center and NASA headquarters that the development was able to ramp up.

Assigning Responsibility

Observation: Responsibility for identifying commonality was assigned.

Branch management placed one of the flight software engineers in charge of the process to identify and eventually develop a common system. Placing responsibility onto an individual or group creates accountability and responsibility for the identification of commonality. If a task is not assigned than there is always the risk that it will be overlooked.

Technical Identification

Observation: A heritage analysis was used to identify which functions should be common, and which should be developed independently.

One of the greatest challenges designers are faced with when developing a life cycle offset common system is the uncertainty of the future system. The FSW system attacked this challenge by conducting a heritage analysis of the flight software systems used on past satellites. The different functions and forms used on each of the satellites were presented to a group of stakeholders. Information from the analysis was then used to educate and give insight into architecture and design decisions.

Passive Identification

Passive identification methods are managerial methods that allow opportunities for commonality to be identified, but are not done for the specific purpose of identifying commonality.

Incentivize cost savings

Observation: Competition created the incentive to initiate the cost saving techniques.

In recent years, there has been an increase in the number of organizations competing for the development of unmanned satellites and satellite flight software. This competition caused managers to seek out cost saving design strategies. Competition is a strong incentive for commonality because it encourages organizations to find new ways to improve the system. Without competition or some other incentive it is easy to become complacent with current systems and methods. Function based organizational also have an incentive to simplify development and maintenance work in the future.

Visibility

Observation: Reorganizing the branches and placing all of the flight software engineers in the same branch gave managers the visibility to identify the opportunity for a common system.

A major contributing factor leading to the identification of commonality was the reorganization that moved all flight software engineers into the same branch. The reorganization enabled the managers to identify the opportunity to develop the systems more efficiently. Before the reorganization there was little to no communication, integration, or management in place between the groups that would allow them to easily share their ideas and methods.

Evaluate Opportunities

Commonality does not always benefit a system, but instead often implies a trade. In addition, commonality often affects the system in complicated ways. It is in this fact that the second task in managing commonality exists: to evaluate opportunities and determine whether or not the opportunities are beneficial. The benefits and penalties to the entire system's life cycle need to be evaluated to determine whether or not the opportunity should be implemented.

Cost Visibility

Observation: Engineers had visibility into the cost effects of design decisions, allowing them to design the architecture with life cycle costs in mind.

One of the advantages of estimating software life cycle costs, compared to hardware, is that the vast majority of all costs are labor, allowing development time to be a good approximation for cost. As a result, when evaluating decisions the stakeholders were able to evaluate which design would be quicker or take less effort to develop. This visibility into the cost structure allowed the developers to create the CFS system with the lowest expected life cycle cost, enabling the easiest and quickest development for future applications. The stakeholders that participated in the deliberation process were experts, having experience across the board in developing, maintaining, and operating a flight software system. The stakeholders were able to estimate the time and effort fairly effectively because they had developed analogous systems in the past and had a good understanding of the tasks necessary to develop the CFS.

Analysis Methods

The stakeholders that conducted the heritage analysis and subsequently created the CFS architecture were attempting to create a FSW system that would be flexible enough to be used on all possible applications because the future systems are uncertain. For example, the CFS gives designers the flexibility to easily substitute CFS applications and test the system on virtually any platform. This flexibility can withstand future uncertainties more easily than an independently designed system.

Uncertainty and flexibility in the system should be considered when evaluating architecture and design decisions. In this case, the architectural decisions were educated by conducting a heritage analysis. However, there are more sophisticated methods of analyzing design decisions, in which flexibility of the system is taken into account. It is for this reason that an economic model for evaluating opportunities for commonality as a real option may be more appropriate than deterministic cost analysis methods. Chapter 7 presents the options-based economic model created to better evaluate opportunities for commonality.

Implement Beneficial Opportunities

The last task for the management of commonality is the successful implementation of opportunities that are determined to be valuable. The implementation task occurs once the opportunities are evaluated and deemed beneficial. Once the decision has been made to implement commonality there were several methods observed to maximize the benefits, including methods to minimize divergence, create flexibility, create traceability, and make the system self-sustaining.

Control Divergence

Observation: Developing the CFS independent of a particular application prevents divergence that could have been caused by a particular satellite's schedule requirements.

It was observed, during the creation of the SAMPEX FSW system that an individual satellite's schedule constraints can force development to take the path of least resistance, independent design, and force the system to diverge. To manage this challenge an independent CFS development flow was created that allowed the CFS to be developed independent of a particular satellite. The independent development flow also allowed portions of the CFS to be deployed as soon as they matured, allowing developers to obtain benefits earlier, shown in Figure 29. For example, the SDO and LRO missions were able to obtain some benefits from the system before development was complete. The independent development stream allowed developers to meet their commonality goals without being forced to compromise development because of schedule constraints.

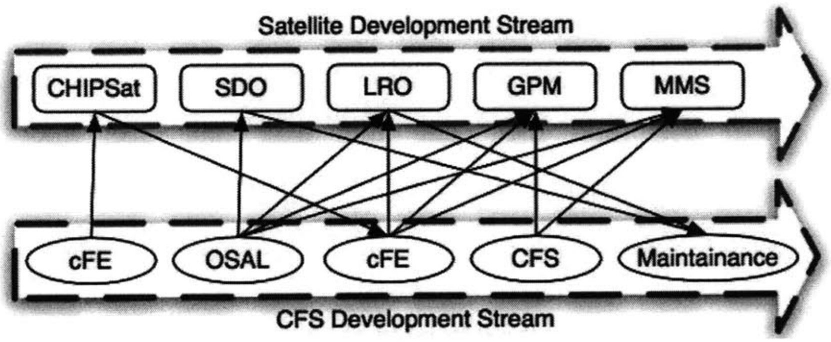


Figure 29: Independent development stream for the CFS system

Flexibility

Observation: The CFS was architected specifically to be flexible enough to be applied to all future satellites.

The CFS architecture was created to offer maximum flexibility, with the specific goals of being scalable, interoperable, and operationally flexible. These goals were achieved in several different ways. First, the system was created with a layered architecture and common interfaces with the satellite specific hardware and the operating system. The OSAL allows the CFS system to be run on almost any operating system, including desktops, which allows scientists and engineers to develop and test prototypes on their own computers. Second, the core of the CFS system, the core Flight Executive (cFE) layer, was designed to be as minimalistic as possible so that it could be used on systems of all sizes while the applications were created with common interfaces, so that the system could be scaled up in size and capability to meet the requirements of any satellite. Finally, the CFS was developed to operate in a run-time environment, adding operational flexibility.

Flexibility ensures that the system can be affordably altered and adapted to meet the requirements of any specific satellite. If a common system is not flexible than it is at risk of having a large integration penalty, which offsets the potential benefits from commonality and at greater risk for divergence.

Traceability/Transparency

Observation: The flight software branch is working to reuse more than just code, but as much of the development and testing work as possible.

There have been significant failures in the past because of the blind reuse of code. These failures include the Ariane V first launch that destructed shortly after lift-off and the Mars Climate Orbiter, which crashed into the surface of Mars (Leveson & Weiss, 2004). Each of these failures was caused by the improper reuse of code. One step that is being used to prevent improper reuse of code is to reuse more than just the code, but also the requirements, documentation, and testing software in the form of a library. Each of the documents in the library should further educate the users and reduce the risk of improper reuse.

The CFS is a large and complicated FSW system and there are an increasing number of users from several different centers. The system may be used more intelligently if there were some sort of training program or class to better educate the potential users about the system and system design. The documentation in the library should have all of the necessary information to reuse the system, but there could be a better way of disseminating that information to others. The CFS developers are looking into other ways to further educate outside users on the use of the CFS.

Self-sustaining

Observation: The CFS continues to improve because it has become self-sustaining.

As more users within GSFC and across NASA begin to use the CFS, the system must be further developed and continually maintained. In order to pay for further development and maintenance, CFS developers have imposed a user "tax" to offset the costs. The CFS users are taxed on the order of one or two Full Time Equivalents (FTE). This tax is relatively small compared to the cost of developing an entire FSW system from scratch and users are getting instant benefits because they are able to have a prototype up and running in a matter of weeks, where it used to take several months. As a result, the CFS is getting more attention and more users, which allows for further development and improvements to the CFS.

5.4 Conclusion

Commonality can greatly benefit a system if it is managed properly. A case study of the Core Flight Software System (CFS) was conducted to determine how commonality is currently being managed within NASA as well as to identify best practices and challenges associated with commonality management.

The creation of the CFS was initiated by a combination of a GSFC reorganization that brought all of the flight software engineers under the same branch management and an increase in competition that led managers to seek more efficient methods of creating flight software. Once it was determined that a new common system should be developed to give the branch a competitive edge, the branch initiated a heritage analysis to evaluate the architectures of satellites previously developed at GSFC. The heritage analysis was used to determine which aspects of the system should be common and which should be unique. The idea was that if a system could be designed to be flexible enough to work on previous systems than it would be flexible enough for future systems.

In a previous attempt at implementing commonality on the SAMPEX system, it was observed that the schedule constraints of the satellite led to compromises to commonality goals. This problem was avoided in the CFS by developing the system independent of a particular application, in a parallel development stream. Portions of the CFS were then deployed on a satellite application as they matured, giving the developers benefits earlier in development. However, independent development can only take place with independent funding.

Portions of the CFS have been launched successfully on three separate satellites, but the entire system is yet to be flown. Until the entire system is flown, it cannot be determined that the system is successful, however early indications show that development has been accelerated and simplified for users. The preliminary success of CFS can be attributed to its unique development, in particular the heritage analysis that was conducted to mature the requirements and architect the system as well as the independent development stream. However, as more groups use the system it is important that information is disseminated to these groups in an efficient manner so that the CFS can be used successfully.

6. Cross-Case Analysis and Management Guidance

The goal in conducting the three NASA case studies was to determine the current practices for the management of commonality within NASA. This chapter summarizes the best practices and key challenges from the NASA case studies, and provides management guidance based on the case study findings.

The chapter begins with a discussion of the commonality trends first discussed in Chapter 2: life cycle offsets and divergence. Next, the different management approaches are discussed, with analysis of both the NASA and industry case studies. Finally, the management methods used to identify, evaluate, and implement commonality are discussed. Guidance on which management method to use is provided, conditional on various system and development scenarios.

6.1 Commonality Trends

In all three NASA case studies and seven industry case studies conducted to date (Boas, 2008), both life cycle offsets and divergence were observed. This section describes how these trends materialized in the case studies, the factors that caused life cycle offsets and divergence, and the effects that they had on the systems development.

6.1.1 Life Cycle Offsets

In each of the seven industry case studies conducted by Boas (2008), it was observed that while planning for each system may take place at the same time in a parallel but coordinated manner, the remainder of the life cycle was offset in time, i.e. the design, manufacturing, and launch of one system occurs later than the first. Similarly, life cycle offsets were also observed in all three NASA case studies.

Life cycle offsets created several challenges for the management of commonality in the NASA case studies and decreased the potential benefits from commonality. A few of the challenges created by offsets in these case studies were:

- Future variants are less certain, causing development decisions to be biased towards the first system
- Potential production and operations benefits due to commonality decrease because less benefit is obtained from learning and economies of scale
- The benefits of commonality are shifted to future systems, while an up-front penalty is placed on the first system; the up-front penalty is especially challenging in the fixed budget scenario seen in NASA case studies

Despite the many problems that offsets create, the existence of offsets is unavoidable in most development situations. Each of the contributing factors that create offsets are universal issues, observed on all industry and NASA development projects, as a result of:

- Market factors - testing market with first variant, or different dates of need
- Technology factors - technology capability development; and learning from earlier products
- Organizational factors - organizational focus on a product, and human resource constraints
- Financial factors - total program cost, and cash flow management, budgetary restrictions

In the Constellation Space Suit System (CSSS) case study offsets were primarily created by the Constellation Program's schedule causing each suit configuration to be required at a different time, i.e. market demand. However, the Constellation Program's schedule itself is primarily offset because of financial and resource constraints, but also to allow learning between the missions. Launching a mission to the Moon before attempting to continue on to Mars decreases the risk of the Mars mission. Thus all four factors causing offsets came into play in this case.

In the Core Flight-Software System (CFS) case study, offsets were also created by the satellite development schedule or market need. Satellites exhibit offsets because of different dates of need. Financial and resource constraints of the development team also came into play.

The International Standard Payload Rack (ISPR) case study exhibited offsets resultant of different dates of need and launch vehicle constraints. The ISPR case study however, was not limited by financial constraints unlike the other case studies because commonality was being implemented between peer organizations. As a result, developers were able to accelerate portions of the development to allow for intelligent negotiations between the peer organizations.

It is interesting to note that offsets were observed in the three NASA case studies, and the causes were all explained by the factors previously identified by Boas (2008).

6.1.2 Divergence

Another trend that was observed in each of the industry case studies, conducted by Boas (2008) was divergence. Divergence is the decrease in commonality between two systems observed as time progresses. Divergence can occur as a result of acceptable factors that improve the systems or as a result of unacceptable factors that do not benefit the systems. Divergence was observed in each of the NASA case studies to occur as a result of:

- Acceptable
    - Technology – New technology has become available that can improve the product if divergence occurs
    - Learning – learning during development showed that a unique solution would either perform better or offer economic benefits
    - Market change – scientist's demand varied between the international partners
- Unacceptable
    - Poor management
    - Intentional pursuit of uniqueness
    - Failure to consider life cycle benefits

Divergence was observed in each of the NASA case studies as well. The initial CSSS architecture (ESR1) contained extensive commonality. However, as development continued more information was obtained on the risks of operating some of the common components in the different mission environments. It was discovered, for example, that having common Body Seal Closures (BSC) resulted in crew injury risk during descent, divergence resulted as this feature was dropped from the design. From NASA's perspective the divergence was caused by learning, however from the contractors perspective there was a requirements change, i.e. a market change.

The ISPR case study presented a method of controlling divergence, but divergence still occurred. Commonality in this case was created in the form of standards that had to be agreed upon by the peer organizations. As a result, any changes from the agreed upon standards would also have to be agreed upon by all of the peer organizations. Divergence did occur, however in the form of waivers to the required standards. Many of the waivers were created as demand for scientific work evolved and additional interfaces were required, i.e. a market change.

The CFS case study also exhibited some divergence. The initial design contained a common device abstraction layer, but as development continued it was found that this layer was not economically beneficial and it was dropped from the design of the CFS, i.e. learning caused divergence.

In the three NASA case studies each of the observed causes for divergence was beneficial to the system or acceptable causes for divergence, indicating that each case study had measures in place to control divergence.

6.1.3 Conclusion

Life cycle offsets and divergence were observed in all of the NASA case studies and were caused by several of the same factors that were observed in the industry case studies. The causes of both life cycle offsets and divergence appear to be universal and likely to appear in most applications of commonality, in industry and government alike.

Table 4 summarizes the observations of both offsets and divergence in the NASA case studies. Life cycle offsets and divergence have significant impacts on system development and should be the considered when managing commonality. Section 6.3 discusses the management methods that should be used to manage the challenges related to the trends.

Table 4: Summary table of life cycle offsets and divergence in the NASA case studies

	Constellation Space Suit System (CSSS)	**International Standard Payload Rack (ISPR)**	**Core Flight-Software System (CFS)**
Life Cycle Offsets -Cause/s	Yes -Constraints -Market -Learning	Yes -Constraints -Market	Yes -Constraints -Market
Divergence -Cause/s	Yes -Learning (technical)	Yes -Market Demand	Yes -Learning (economic)

6.2 Management Approaches

There are several different approaches to managing commonality within systems, first described in Chapter 2. In the three NASA case studies and seven industry case studies (Boas, 2008) three broad approaches for applying and managing commonality were observed: (1) reactive reuse of previously developed components, (2) the common building block approach, and (3) a process for identifying, evaluating, and implementing wide-spread forward commonality. The extremes of these management approaches are development with absolutely no commonality and the development in commonality culture, as shown in Figure 30. Each of these approaches is described in detail in this section in relation to the observations in the NASA case studies, including the challenges and methods that proved beneficial for each approach.

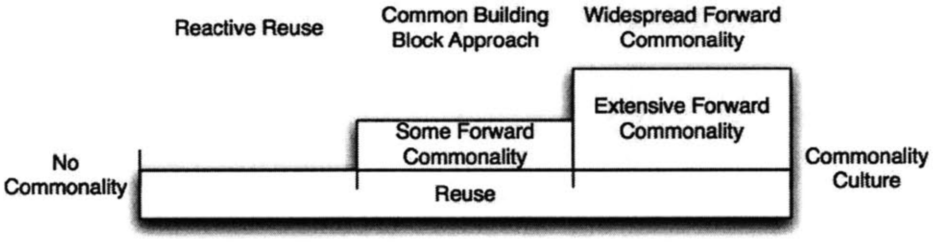


Figure 30: Three high-level approaches for the management of commonality with the extremes of no commonality and the creation of a commonality culture

6.2.1 Reactive Reuse

The reactive reuse approach is characterized by a development focus on a single system at a time with little to no long-term planning, integration, or identification of commonality. However, there is some effort placed on reusing components that were developed for previous system. Without any forward planning or integration the amount of reuse, and as a result the benefits from reuse are limited. However, reactive reuse is still the sole approach pursued by some managers because there is no up-front development penalty.

The reactive reuse approach to managing commonality was observed to be the lowest level of commonality management because no effort is placed on forward planning or increasing the amount of potential commonality. Instead, developers attempt to reuse what has already been developed. The most successful example of reactive reuse was observed in the industry case studies in which the previous system was used as the baseline for the next system (Boas, 2008).

The systems chosen for study were picked because they exhibited some level of forward commonality, therefore none of our case studies exhibited the reactive reuse approach. However, the reactive reuse approach has been used extensively by NASA as seen in the reuse of the J-2X rocket engines from Apollo on the Ares I and Ares V launch vehicles and the reuse of the Solid Rocket Boosters (SRB) from the Space Shuttle also on the Ares I and Ares V launch vehicles. Reactive reuse cannot be applied broadly across the system unless the variants are extremely similar to each other and is usually associated with a relatively large integration penalty.

6.2.2 Common Building Block Approach

The next management approach observed was the common building block approach. The common building block approach is characterized with the development still focused predominately on a single system at a time, however some effort is placed on the development of forward commonality on a few high value components. The common building block approach still contains processes to identify, evaluate, and implement commonality, however the identification and evaluation is limited in scope.

When offsets exist in a system there is a tendency to bias development towards the first system. In order to ensure that development is not biased towards any individual application in this approach an independent development process should be used, that works closely with the management of both systems.

This approach was observed in both the International Standard Payload Rack (ISPR) case study and to some extent the CFS case study. The laboratories developed for the ISS by NASA, ESA, and CSA, were each developed predominately independent from each other. However, the architecture of the ISS and the utilization agreement between each of the partners encouraged them to focus on the creation of commonality in a few high-value components, including a common payload management system. The Interface Control Working Group (ICWG) was then created to manage the interfaces of the ISPR and enable the international partners to negotiate and agree upon a common design for the systems.

From the high-level satellite point of view the CFS is a high-value component in which effort was placed to develop commonality, while satellites are still developed independently. However, from the FSW point of view the system was developed through a process to apply widespread forward commonality, by managing the *identification* of opportunities, *evaluating* the opportunities, and finally *implementing* the beneficial opportunities. Therefore, the details of the CFS management methods will be discussed in more detail as an example of widespread forward commonality.

6.2.3 Widespread Forward Commonality

The third approach observed to manage commonality was the widespread identification, evaluation and implementation of forward commonality. This approach involves significant effort to seek out and *identify* opportunities for commonality across the entire system. The opportunities are then *evaluated* to determine the benefits and penalties, and finally the opportunities are *implemented* if deemed beneficial. This approach for commonality management was observed in both the CFS and CSSS case studies.

Commonality was managed in the CSSS case study through a formalized iterative process. This process began with the identification of opportunities, in which the original system architects evaluated each expected Contract End Item (CEI) to determine whether or not the CEI could feasibly be common. If it was determined that it could become common, the CEI was baselined as such. Thereafter, any proposed changes to the architecture must be evaluated by the architecture and analysis group and approved in a multi-tier approval process.

Commonality was also managed in the CFS case study through a formalized process. The process began with management analyzing previous systems to identify the extent that the systems could be reused. It was determined that the maximum benefits would be obtained from a new system, created with the intent that it be flexible enough to be reused on each of the future missions. Design decisions were analyzed and deliberated upon by experienced engineers, with insight from a heritage analysis. The system was deployed in stages and continues to be maintained and further developed as needed by an application independent group with the intent that it continues to be used on all future satellites.

The widespread forward commonality approach was observed in both cases to be managed through a formalized process, but Boas suggests that the ideal scenario involves a "commonality culture" in which commonality becomes so engrained in the organizations culture that the formal processes become less important. The NASA case studies showed that while the formalized process was integral to the management of commonality, several other opportunities were identified by experienced engineers that were enabled to seek out commonality. Therefore, it can be shown that the commonality evolves from this approach, by giving engineers the ability and incentive to seek out additional opportunities independent from the formal process.

6.2.4 Conclusion

Each of the management approaches described above have pros and cons. While the reactive reuse method does not have an up-front development penalty, the approach is limited in scope and as a result can only obtain limited benefits. The common building block approach allows management to obtain a large portion of the benefits from forward commonality, however without the consistent and widespread evaluation of opportunities potential benefits of commonality may be overlooked. The widespread forward commonality approach allows for the maximum life cycle cost benefits, however these benefits come at the cost of an up-front development penalty. Managers must then decide which approach is best suited for their system. Table 5 summarizes each of the management approaches as observed in the NASA case studies.

Table 5: Summary of each of the management approaches observed in the NASA case studies

	Forward Commonality	**Management Approach**	**Benefits**	**Problems**
Reactive Reuse	No forward planning or commonality	Reuse previously developed components	No up-front development penalty on the first system developed	Commonality is greatly limited in scope compared to forward commonality approaches
Common Building Block	Forward commonality is applied to a few components	Identify high value components with potential to be common	Obtain benefits from commonality on high-value items	Potential to overlook significant benefits from commonality
Widespread Forward Commonality	Widespread search for forward commonality	Processes to identify, evaluate, and implement opportunities for commonality across the system	Maximizes the potential benefits from commonality while avoiding unbeneficial opportunities	Widespread identification of commonality creates an up-front development penalty

6.3 Management Methods

In order to implement forward commonality in either the common building block or widespread forward commonality approaches successfully three fundamental tasks are required: (1) to manage the *identification* of opportunities for commonality, (2) *evaluate* opportunities to determine whether or not they are beneficial, and (3) to *implement* opportunities for commonality that are deemed beneficial. The following section describes the lessons learned and management guidance obtained from the NASA case studies in regards to the fundamental management tasks and characteristics of the system being developed, including: (1) development method, (2) organizational structure, (3) system type, and (4) the relative capability and production size of the systems.

6.3.1 Manage the Identification of Opportunities

The initial identification of opportunities for commonality is arguably the most important and the most challenging task, because the task is greatly complicated by life cycle offsets. This section presents the challenges related to identifying opportunities for commonality and guidance on how to best manage the task. There are four primary challenges in the identification phase, related to: timing, uncertainty, the up-front investment, and scope of application. Each of these challenges is discussed below with guidance based on the case study findings.

Timing

The first challenge that was observed in each of the case studies, is that the ability to affect the systems costs and the potential benefits from commonality decrease as development progresses over time.

Guidance 1: The identification process must begin early in development in order to maximize the potential benefits.

The identification of potential forward common components should take place early in the development process in order to maximize the potential benefits from commonality. As time progresses the ability to influence development and costs decreases, therefore the timing in which commonality is identified is important in determining the potential benefits. Commonality was identified early in development in both the CSSS and CFS case study, enabling the application of widespread forward commonality.

Uncertainty

Offsets present several challenges to the identification stage, with one of the greatest challenges being the uncertainty of the future system. The NASA case studies presented several methods that can be used to manage the uncertainty of future systems.

Guidance 2: Design with flexibility to reduce the effect of uncertainty on the system.

Designing a system to be flexible offers two benefits to common systems. First, it allows uncertainty of future systems to have a smaller effect on the success of commonality because the system is able to adapt to any potential changes. As a result, the likelihood that divergence will occur decreases and opportunities that were identified are more likely to be implemented. Second, the cost of integrating intended common components into future systems is lower if they are designed to be flexible. Flexibility was designed into the systems of each NASA case study and open architecture design techniques were also implemented in the CSSS case study to offer development flexibility.

Guidance 3: Reduce uncertainty in future systems by analyzing trends in the applicable previous systems.

It may be possible to reduce uncertainty in future systems by analyzing trends in the applicable previous systems. By analyzing how past systems have evolved or trends that have occurred in development it may be possible to predict and manage uncertainties in the current system. In the CFS case study the decision makers were able to gain insight on architecture design decisions by analyzing previously developed systems in what they referred to as a heritage analysis. The basic idea was that if a system could be developed with the flexibility to meet the previous systems requirements than it is more likely to be flexible enough to meet future system requirements. This method is most useful when there are a relatively large number of similar previous systems.

Guidance 4: Accelerate portions of the design to better inform the identification process.

In cases that two common systems are being developed and there is substantial uncertainty in the future system there may be tests or development tasks that can accelerate or mature development of the future system. In the ISPR case study standards were implemented between peer organizations. As a result, the standards had to be agreed upon by all parties. In order to better inform decisions and negotiation between the peer organizations portions of development were accelerated to the extent that they were done nearly in parallel. There are several factors that often prohibit parallel development including financial constraints, but this restriction did not apply to development between the peer organizations.

Up-Front Investment

Developing forward commonality into a system in which there are offsets implies an up-front development penalty while offsetting benefits to future systems. This up-front development penalty creates a tremendous hurdle for management because the future benefits are often as uncertain as the future system. The hurdle is even greater in the government because managers are forced to work within fixed budget constraints.

Guidance 5: The first key to overcoming the up-front investment is support from management.

An up-front effort must be made in order to seek out opportunities for commonality and to eventually implement those opportunities. Developers will not willingly put themselves at risk of going over budget or overdue in order to apply commonality. Therefore management support is needed in order to stimulate the identification process. Strong management support for the development of commonality was observed in all three of the NASA case studies.

Guidance 6: Incentives can be used to encourage developers to make the up-front investment.

An additional method that was observed to help overcome the up-front development penalty was the use of incentives. An incentive for commonality can be put in place to offset the risks associated with the up-front investment. A function-based organizational structure creates a natural incentivize for commonality because developers obtain workload benefits from commonality. In the CSSS case study managers incentivized commonality by directly correlating the size of the award fee to the ability to control costs and specifically develop and implement a commonality plan.

Guidance 7: Assign responsibility for the identification of commonality.

Responsibility for the identification of opportunities for commonality should be assigned to a person or group. If a specific person or group is not assigned responsibility for seeking out opportunities for commonality the job will often get overlooked or dropped to the bottom of the priority list. In both the CSSS and CFS case studies assignment of this responsibility was observed to be essential to the identification process.

Scope

To obtain the maximum benefits from commonality opportunities for commonality should be sought out across the system on a wide scope.

Guidance 8: The extent to which an individual or group can identify opportunities for commonality is limited by their visibility.

Opportunities for commonality can be identified solely by individuals or groups that have the visibility to see the opportunity. Therefore, any person or group that is expected to identify opportunities should be given visibility across the systems. Boas describes the ideal management approach to be one in which the identification, evaluation, and implementation of commonality becomes engrained into the corporate culture to the extent that a formal process becomes less important. In order to approach this ideal the engineers within the organization must be enabled to identify opportunities, and therefore they must be given the visibility to see opportunities. In the CSSS case study visibility was extended through the design of the organizational structure and the creation of integration groups.

6.3.2 Evaluate Opportunities

Not all opportunities for commonality are beneficial. In fact commonality almost always presents some sort of trade between competing goals. It is essential that each opportunity that is identified be evaluated to determine the benefits and penalties of the opportunity. However, in all three NASA case studies and seven industry case studies (Boas, 2008) managers were observed to lack the tools necessary to quantitatively evaluate the effects of commonality on the system. This is one area that was observed to need significant improvement. Chapter 7 will discuss the work done to improve the economic evaluation of commonality. Despite the shortcomings observed in the evaluation process there were important lessons obtained from the case studies.

Guidance 9: Evaluate monetary and non-monetary benefits and penalties.

Commonality implies a trade between competing goals. This trade often involves monetary benefits while penalizing system performance. Therefore, it is important that both monetary and non-monetary benefits are analyzed before a decision to either implement or not implement commonality is made.

Guidance 10: Give engineers visibility into the monetary effects of decisions and they will use that information to lower the system costs.

Engineers and management will use information that they have in order to make decisions. If they have visibility or information on the economic impacts of different designs the information will be used to make the best system. One of the advantages of developing commonality into a software system is that software has a simple cost structure almost completely composed of labor. Experienced engineers have visibility into the economic consequences of design decisions because of the simple cost structure, enabling them to develop a system with the lowest expected life cycle costs. If engineers are not provided information on the life cycle cost impacts of designs, than that information will not be used to make decisions.

6.3.3 Implement Beneficial Opportunities

Once an opportunity for commonality has been identified and determined to be beneficial it should be implemented in a way that the maximum benefits are obtained. The greatest challenge for the implementation phase observed in the NASA case studies is controlling divergence. Divergence is not always bad, but unacceptable causes for divergence should be eliminated.

Guidance 11: Formal control processes to review and approve divergence will reduce unacceptable causes for divergence.

To maximize the benefits from commonality unacceptable causes for divergence should be controlled. A formalized control process can be used to eliminate the unacceptable causes. A formal process was observed in the CSSS case study to control changes and divergence to the system. The control process involves a trade study followed by a multi-tier approval process. The control process effectively ensures that when divergence occurs it is for an acceptable reason. In the ISPR case study the formal control process was created through the use of standards. Each of the international partners must agree on any changes to the already approved standards. The CFS is maintained and further developed by an application independent group with the goal of maintaining the maximum forward commonality.

Guidance 12: Design with flexibility to allow a system to adapt to unanticipated changes.

Uncertainty in the future system can eventually result in divergence if the system is unable to adapt to the system changes. Designing the system to be flexible allows the system to change with a smaller integration penalty, preserving the benefits from commonality. All three of the NASA case studies were designed to be flexible.

Guidance 13: A development stream independent of a particular application ensures that development is not biased towards a specific application.

When commonality is being implemented into a system there is the possibility that development will be biased towards a specific application. If a single system is responsible for developing forward commonality between multiple systems, but is only responsible for the first system then the project will tend to be biased towards the first system. An independent development group is able to develop the system without bias towards one project or the other. The CFS system was developed independent of a particular satellite application in order to ensure that commonality goals were not compromised. The CFS developers were also able to obtain benefits sooner by staging deployment. The function-based organizational structure of the CFS enabled the creation of an independent development flow.

6.3.4 Impacts of System Characteristics

Not all systems or development scenarios should be managed the same way. There are several common management techniques that can be applied universally, but while conducting the NASA case studies several factors were identified that affect the management method. The factors include: the development method, the organizational structure, the system type, and the production quantity and capability of the system. Each of the factors is discussed below along with management guidance.

Development Method

Two separate methods of development were observed in the NASA case studies: (1) in-house development and (2) external or contracted development. All of the industry case studies exhibited in-house development, while NASA often contracts out development of its systems. External development and the contracting structure that NASA is required to use make it difficult to manage commonality. However, the case studies have provided some guidance.

Guidance 14: Intelligent contract writing can be used to incentivize the contractor to seek out and implement commonality when beneficial.

Intelligent contract writing was observed to be useful for the management of commonality being developed by a contractor. In the CSSS case study the contract was chosen to be a cost-plus award fee contract in which the award fee is directly related to the ability of the contractor to control life cycle costs, with specific attention given to the development and application of a plan for commonality.

Guidance 15: Contract writing can be used to avoid potential proprietary restrictions that have been witnessed in the past.

The managers of the CSSS integrated the development of the configuration 2 suit as an option in the contract, allowing management to choose to practice the option for development if they are satisfied with current performance or choose to seek competitive bids if they are not satisfied. However, the threat to seek competitive bids will only prove valuable if it is legitimate. In previous development projects contractors have been able to hold onto the intellectual property of vital parts of the system, forcing management to use the same contractor on future development. The CSSS contract specifies in detail all aspects of development that are to be handed over to NASA, ensuring development flexibility.

Organizational Structure

In nearly all case studies, NASA and industry alike, the organizational structure consisted of a single organization applying commonality within its own development projects. However, the ISPR case study presented a case in which commonality was being developed between multiple peer organizations, in which there was no central management in place between them. As a result, managing commonality between the systems is more challenging.

Guidance 16: When managing commonality between peer organizations, utilize standards.

In order to manage commonality between NASA and its international partners, commonality was implemented in the form of standards. The standards were agreed upon by each of the international partners. After the agreement, any changes to the system must be negotiated and again agreed upon. Mutual agreement is the simplest and most structured way of implementing commonality between peers. Standards are traditionally implemented at low levels of design, such as the component level, however entire system interfaces and portions of the laboratory design were implemented successfully in the form of standards.

System Type

All of the industry case studies, conducted by Boas (2008), evaluated systems primarily composed of hardware. In order to gain a broader perspective on how the system type affects the management of commonality, a software system (CFS) and a service system (ISPR) were also studied. There are several differences that should be considered when managing commonality.

Guidance 17: Reuse of software systems should be carefully tracked and tested.

The internal interactions between hardware components can usually be easily analyzed and tested. However, there have been several failures caused by unpredicted interactions with reused software, including the Ariane V first launch failure. To manage this danger software should be closely tracked and carefully tested for each reuse. Even if a software system is extremely reliable it does not mean that it is necessarily safe (Leveson N. , System safety, 2008). To manage the cost of the testing the CFS developers also created a test suite that can be reused for each application.

Guidance 18: Software costs are more visible to developers and allow developers to consider economic consequences of designs.

One of the benefits of developing software systems is the greater cost visibility inherent in software development. The cost structure of software is completely composed of labor, which makes economic evaluation simpler. In development of the CFS, the simple cost structure enabled developers to make development decisions that would reduce life cycle costs. If engineers do not have visibility in to the cost structure than that information will not be used to make design decisions.

Guidance 19: Service systems or other systems that are highly subject to customer demands should be flexible enough to adapt to changing needs.

Service systems provide a service to a customer or customers. Customer's demands have the tendency to evolve over time, the system should have the ability to evolve with the changing demands. The original plan for ISS experimentation was to install the experiments into ISPRs and continuously transport them to and from the ISS with an MPLM. However, the plans for shuttle operations and the demand for experiment size have both changed. As a result, several EXPRESS racks were built that offer additional flexibility in the size of payloads and the transportation method for those payloads.

System Capability and Production Size

In a set of common systems one of the systems will often be more capable than the other and one will often have a larger production volume than the other. This section will discuss how these differences impact the life cycle cost benefits.

Guidance 20: To maximize the life cycle benefits from commonality, create the larger volume and more capable system first.

The last factor that should be considered when developing common systems is the order in which to develop these systems. It is not always possible to choose the order of development, however the ability to choose can add value to the systems. While the affects of commonality on the development phase are fairly straightforward, the affects on production and operations are less obvious. The benefits and penalties related to the production and operations phases are related to the system capability and production volume. The specific formulas and implementation of these benefits and penalties are discussed in detail in Chapter 7.

From the point of view of system A, it is advantageous to develop the more capable system first, because there is a smaller *excess capability penalty* (p_{capA}) on system A. A component that is developed to meet additional requirements is assumed to be more complex than independently developed components and as a result is more expensive to produce and operate. The added expense caused by this complexity is described as the *excess capability penalty* (p_{cap}). It is also beneficial to system A, to have the system with the larger production volume developed second. Having the larger production system developed second allows system A to obtain the maximum *learning* and *economies of scale* benefits during any overlapping production and operations phases. Figure 31 shows the best possible development order for system A.

However, the order of development that maximizes the benefits for system A do not necessarily maximize the benefits for system B. Producing the more capable system second reduces the excess capability penalty on system B. In addition, system B is most benefitted if system A, before it, has a larger production size; allowing system B to obtain the maximum production and operations phase benefits, as shown by Figure 31.

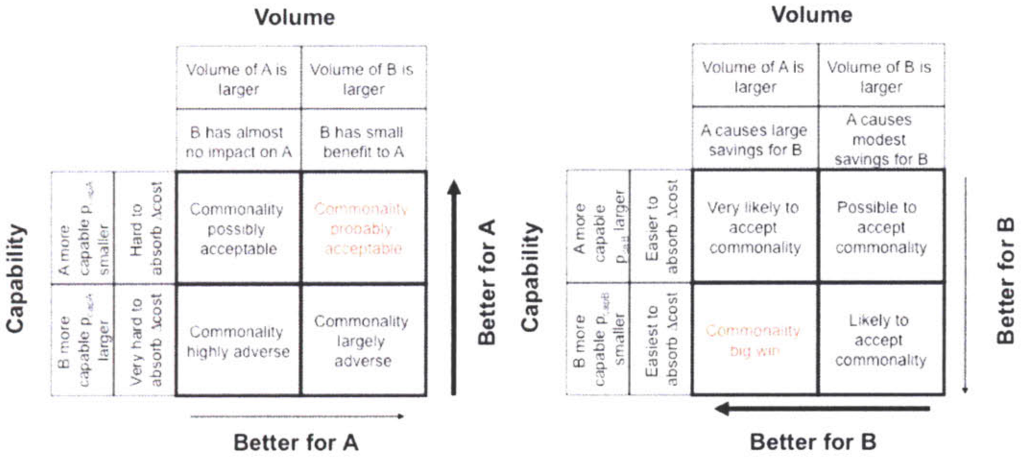


Figure 31: Benefits to individual systems based on order of capability and production volume

If the decision maker for commonality is also the manager of either system A or system B than decisions for the system may be biased towards one system or another. Neither of the described plans will maximize the benefits for the life cycle of both system A and B. The maximum benefit to the life cycle is provided when the first system, system A, has the larger production volume of the two, providing substantial production and operations benefits to B. In addition, the combination is most benefitted if the more capable of the two systems is developed first, allowing benefits to be provided earlier, as shown in Figure 32.

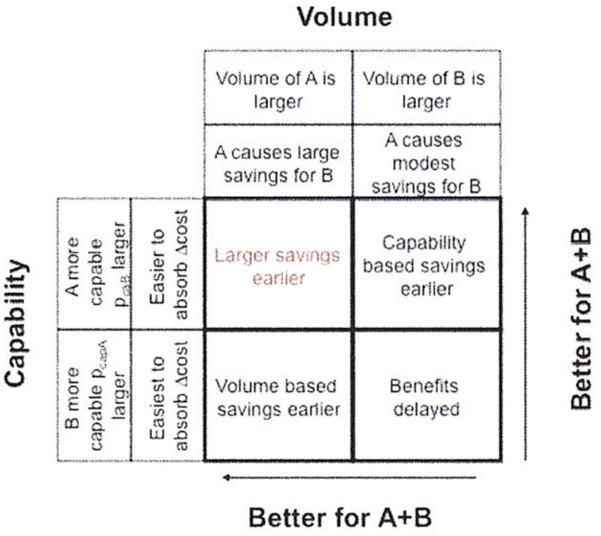


Figure 32: Diagram outlines the best order of development in order to maximize the life cycle commonality benefits

6.4 Conclusion

The goal of this research was to determine how commonality should be managed within NASA systems and what the differences are between NASA and industry applications of commonality. To that end, three case studies were conducted on past and present NASA systems, each of which was chosen to present a diverse view of system types, development methods, and organizational structures. Despite the multiple differences between the NASA case studies and the industry case studies several similarities were also observed:

- Life cycle offsets and divergence appear to be universal concepts that exist in some form in every development project with commonality. The trends are created by factors that were consistently observed in NASA and industry alike.

- There appears to be a wide spectrum of approaches for the management of commonality with differing amounts of effort placed on forward commonality and extremes of no commonality and the creation of a commonality culture, as seen in Figure 30.

- A set of critical management tasks required to apply forward commonality to a system exist. These task include managing the *identification* of opportunities, *evaluating* the opportunities, and *implementing* those opportunities that are deemed beneficial. For these management tasks there appears to be a common set of good practices that can be used to maximize the benefits from commonality, such as strong management support, assigning responsibility, increasing visibility, and designing with flexibility. The management methods are explained in detail in section 6.3.

There are also several differences between each system that should be considered when managing commonality. Such as the type of systems being developed, the organizational structure, the development method, and the production volume and capability of the common variants. Table 6 summarizes each of the key aspects of the NASA case studies. The table also shows the most common management methods observed in the industry case studies, but does not discuss details of any of the individual industry case studies.

Table 6: Summary of the management findings from the three NASA case studies and the seven industry case studies

	Constellation Space Suit System (CSSS)	International Standard Payload Rack (ISPR)	Core Flight-Software System (CFS)	Previous Industry Cases
Program	Constellation Program	International Space Station	Unmanned Scientific Spacecraft	4 Aerosapce, 3 Others
Product Type	Hardware	Service	Software	Hardware
Development Type	Contracted/External	In-house	In-house	In-house development
Organizational Structure	Single Project	International Peer Organizations	Single Project	Single Project
Life Cycle Offsets	Yes	Yes	Yes	Yes
Divergence	Yes	Yes	Yes	Yes
Management Approach	Widespread Forward Commonality	Common Building Block	Widespread Forward Commonality	Prodominately Reactive Reuse
Culture	Family, two configurations for the same common suit	A single standard system to be reused	A common core with common application modules to be continuously maintained and reused	Single-product culture, with large attempts to transition to forward commonality
Identify				
Pursuit of Forward Commonality	Yes, goal was to create two highly common configurations of a single suit system	Yes, goal was to create international standardization of payload systems	Yes, goal was to create a flight software system that can be applied to all systems	Mixed effort for forward comon
Identification Method	Analyzed each expected CEI and baselined if feasible	Identified as a result of utilization rights	Identification resulted from increased competition for satellite development	Previous system was baseline for future development
Management Methods	- Timing - Program Support - Assign Responsibility - Increased Visibility - Incentives	- Timing	- Management Support - Assign Responsibility - Heritage Analysis - Increased Visibility - Incentives	Varied
Evaluate				
Awareness of potential commonality benefits	Yes	Yes	Yes	Yes
Formal analysis of commonality opportunities	Yes, trade studies were conducted for each potential change. Economic analysis was limited.	Limited, goal was to efficiently utilize payloads on international laboratories with limited logistics onground and inspace	Limited, heritage analysis was used to inform decisions made by the deliberation of experts. Economic analysis was qualitatively estimated based on time.	Limited
Implement				
Formal management of commonality	Yes, maximize commonality in ininitial architecture and control divergence	Yes, negotiation and formalization of standards	Yes, CFS is continuously updated and maintained from feedback for each satellite	Informal management
Formal owner of commonality	Yes, Architecture and Analysis Group	Yes, Interface Control Working Group	Yes, CFS development team	Usually at executive level
Control Process for Divergence	Mult-tier approval process for changes to the baseline architecture	All changes must be presented and negotiated at ICWG	CFS is maintained by an independent development group	Limited
Management Methods	- Flexibility - Open Architecture - Standardization - Avoid Proprietary Restrictions	- Flexibility - Compromise - Standardization - Accelerated Development	- Flexibility - Traceability - Self-sustaining	Varied

7. Economic Evaluation of Commonality

Commonality is not an end in itself, but a means to an end because commonality does not always benefit a system, but often implies a trade between competing goals. Therefore, opportunities for commonality must be evaluated, in order to determine whether or not the opportunity should be implemented. One of the weaknesses in the management process observed in the NASA and industry case studies (Boas, 2008) was the evaluation of opportunities to determine the potential benefits and penalties that the opportunity would have on the system. The evaluation process observed in the NASA and the industry case studies was commonly either not conducted or limited to qualitative analysis, with one exception being the military aircraft case (Boas, 2008). The limited evaluation was often a result of either, (1) a lack of knowledge of the specific benefits and penalties of commonality or (2) the absence of tools to assist with the evaluation. This chapter will address these issues by presenting two methods of quantitatively evaluating the specific economic benefits and penalties of commonality. Non-monetary effects of commonality tend to be case specific and are not incorporated in the economic model, but are discussed in section 2.4.

The chapter begins by reviewing the classification of components within a common system, first presented in Chapter 2. The classification of common components forms the basis of the economic model. Next, a deterministic cost model is presented with a specific focus on how the effects of commonality are incorporated into the model. The deterministic cost model is based on work previously conducted by Boas on the evaluation of commonality (Boas, 2008). The cost model is further elaborated on by applying the model to an example from the Constellation Space Suit System (CSSS) case study. Next, an alternative evaluation method is presented in which commonality is evaluated as a Real Option[2] in the system. The cost model to evaluate commonality as a real option builds on the deterministic cost model by additionally considering future uncertainties in the system and flexibility available to management. Finally, the options-based economic model is also applied to the CSSS.

7.1 Classification of Commonality

The existence of offsets and divergence create a necessary decision at the time that the first system (system A) is developed: to either develop components with the intent that they become common with future systems or with the intent that they are unique to system A, as shown in Figure 33.

Designing a component or asset with the intent that it becomes common implies additional requirements and as a result additional cost to develop, i.e. the *cost of common development.* At the time that the future system (system B) is developed, the components from system A either become common between the two systems, or they become unique to system A. Components from systems A and B, can then be classified into five different categories, as seen in Figure 33. Classes 1-4 compose system A, classes 3-5 compose system B, and classes 3 and 4 are common to both systems.

---

[2] "A real option is a technical element embedded initially into a design that gives the right but not the obligation to decision makers to react to uncertain conditions." Therefore, the ability to utilize components that are developed with intended commonality is a real option (de Weck, O., de Neufville, R., & Chaize, M., 2004).

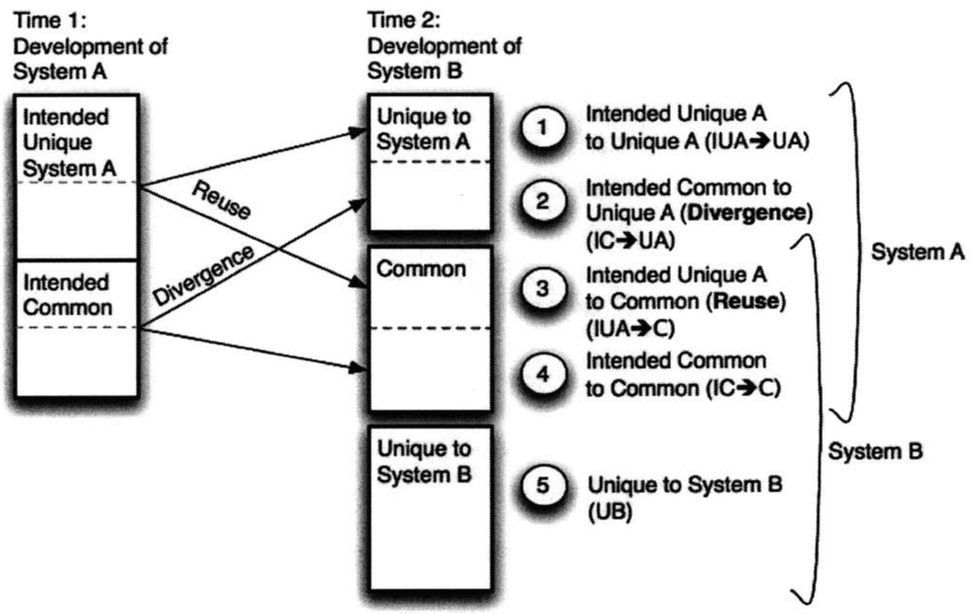


Figure 33: Classification of components within common systems

7.2 Deterministic Cost Model

It is useful to divide the system into the five classes in Figure 33 because each of the classes affects the cost of the systems in different ways. The following section will explain how each of the five classes affects the systems and how knowledge of the affects was used to create a deterministic cost model.

7.2.1 Cost Model Factors

The economic models developed are not stand-alone cost models, but instead a way of updating cost estimates created for independently developed systems. Both, the deterministic and options-based cost models update the cost estimates by explicitly considering the effects of commonality on the entire life cycle. As a result, the model requires several inputs related to the application of commonality in the system. There are two types of inputs required: (1) independent development cost estimates for the entire life cycle of the system and (2) model inputs based on the specific effects of commonality on the system costs, as shown in Figure 34. The next section will discuss all inputs required.

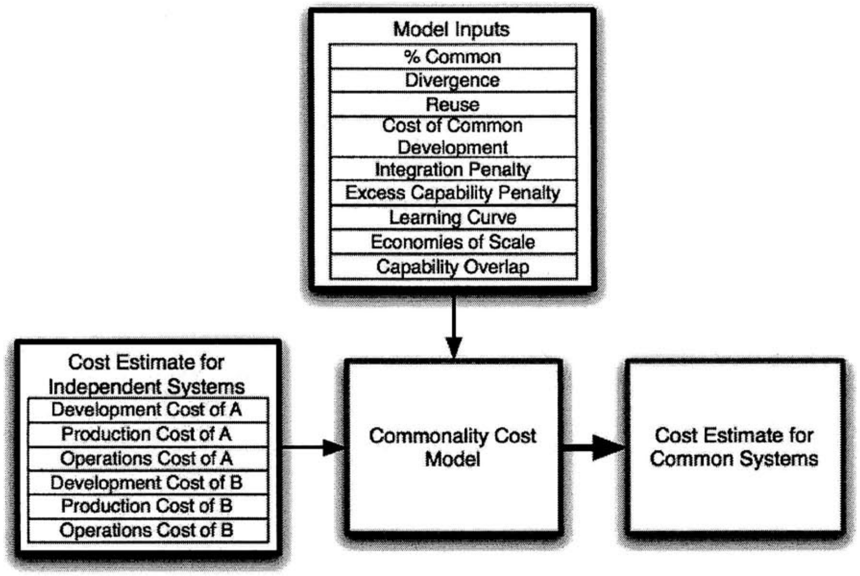


Figure 34: The cost model is not a stand-alone model, but a method of updating independent cost estimates

Independent Development Cost Inputs

The first set of inputs required is the independent development cost estimates for the systems, including estimates for the development, production, and operations phases of both systems. These cost estimates are best made by traditional cost estimation methods, such as parametric cost models and cost estimation by analogy.

Model Inputs

In addition to the independent development cost estimates, several other model inputs are required to determine the affect that commonality has on the cost of the systems. The first three factors required to quantitatively evaluate the financial benefits and penalties of commonality in the systems are: the percent of intended commonality, divergence, and unintended reuse. These factors dictate the relative size of each of the classes in the model.

- *Commonality (C)* – the percentage of system A that is developed with the intent that it becomes common with system B. This factor should be known at the time that the first system is developed and should be quantified.
- *Divergence* (d_{iv}) – the percentage of the intended common components of A that actually become unique to system A. Divergence may be more difficult to estimate because divergence will not have occurred yet and management will attempt to control divergence if possible.
- *Reuse* (r_{iu}) – the percentage of the intended unique components of A that actually become common between the two systems. Reuse may also be difficult to estimate because if management believed that a component could be reused than it would be marked as common.

By quantifying each of the above values the classes can be quantified into a cost basis, using the equations in Table 7. The formulas for the cost basis divide the cost development estimates for the independently developed systems into five categories, while automatically reducing the development cost of system B as a result of the commonality. By dividing the systems into these five classes, the independent cost estimates can be transformed into cost estimates for the common system. The following subsections on the model inputs will discuss each of the classes in relation to how commonality affects development, production, and operations of the systems.

Table 7: List of equations used to quantify the relative size of each of the five commonality classes

Class	Class Description	Cost Basis for Each Class
1	Intended Unique A to Unique A (IUA→UA)	$A*(1-C)*(1-r_{iu})$
2	Intended Common to Unique A (IC→UA)	$A*C*d_{iv}$
3	Intended Unique A to Common (IUA→C)	$A*(1-C)*r_{iu}$
4	Intended Common to Common (IC→C)	$A*C*(1-d_{iv})$
5	Unique to B (UB)	$B-A*(C*(1-d_{iv})+(1-C)*r_{iu})$
Key	A - cost of developing independent A B - cost of developing independent B d_{iv} – divergence of intended common components	C - intended common % of A r_{iu} - reuse of intended unique

Although the cost basis for each class is based on the independent development cost of the two common systems, the class bases are also used to quantitatively evaluate both production and operations costs. This can be done by representing production and operations costs as some fraction of development. For example, many cost models assume that the first unit of production will cost 10% of the total development (Hofstetter, 2009).

The discussion of the classes assumes that system A is developed first, followed by system B after some offset, as shown in Figure 35. In reality, there may be more than one future system, however this assumption allows for a simpler discussion of the economic model. The economic model is capable of evaluating multiple products if extended.

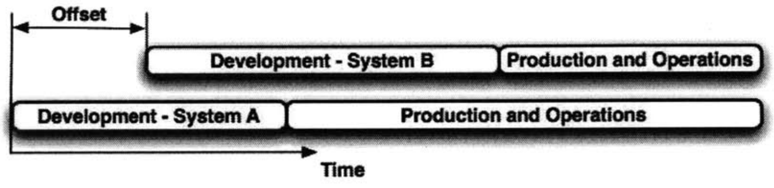


Figure 35: Assumed timeline of development, production, and operations used for the discussion of the effects of each commonality class

Class 1: Intended Unique A to Unique A

The life cycle costs of items that are developed with the intent that they are unique to system A and then become unique are unaffected by commonality. As a result, the cost of this class is best estimated with traditional cost estimation tools, such as parametric cost models or analogy. The cost estimates for this fraction of the system feed directly through the cost model without adjustment, as demonstrated by Table 8 in which a '1' indicates that the cost basis is not affected by a benefit or penalty related to commonality and a '0' indicates that the cost basis is not included in the cost estimate for that phase of the system.

Table 8: Chart describing the factors that influence the cost basis for each phase of the life cycle for each system; a '1' indicates that the cost basis is unaffected while a '0 indicates that the basis is not included in the common system cost estimate

Class		**Development**		**Production**		**Operations**	
#	**Desc.**	**Sys. A**	**Sys. B**	**Sys. A**	**Sys. B**	**Sys. A**	**Sys. B**
1	IUA→UA	1	0	1	0	1	0

Class 2: Intended Common to Unique A (Divergence)

Development

Developing an item with the intent that it becomes common requires the developers to develop the component to meet the requirements of the current system and the requirements of the expected future systems. As a result, the development cost of the intended common components in system A are higher, increased by the penalty *cost of common development* (p_{cd}). The cost of common development represents the additional labor and coordination cost of design and development of the components that are expected to be common with system B. However, divergence occurs, causing the elements in class 2 to be unique to system A. Therefore, class 2 penalizes system A and is not a part of the cost estimate for system B, as indicated by Table 9.

Production and Operations

A component that is developed to meet additional requirements is assumed to be more complex than independently developed components and as a result is more expensive to produce and operate. The added expense on system A is described as the *excess capability penalty (p_{cap})*. The excess capability penalty likely does not affect the different systems and life cycle phases equally, therefore the excess capability penalty is unique to the phase and system in which it is applied. As a result, Table 9 shows p_{capAp} for the capability penalty for system A in production and p_{capAo} for system A in operations. In class 2, system B does not have an excess capability penalty because divergence occurs, causing the components to be developed independently.

Table 9: Table representing the factors that influence the cost basis for class 2

Class		**Development**		**Production**		**Operations**	
#	**Desc.**	**Sys. A**	**Sys. B**	**Sys. A**	**Sys. B**	**Sys. A**	**Sys. B**
2	IC →UA	p_{cd}	0	p_{capAp}	0	p_{capAo}	0

Class 3: Intended Unique A to Common (Reuse)

Development

In class 3, components are developed with the intent that they are unique to system A. Despite this intent the components in class 3 are reused on system B. As a result, system B obtains development benefits from commonality without the upfront *cost of common development* on system A. The reuse will benefit the development phase of system B by decreasing the development work required, however an *integration penalty (p_{int})* will likely be created by the development work required to integrate the previously developed components into system B.

Production

This class of components is not developed with the intent that they become common, as a result system B is benefitted without an *excess capability penalty* on system A. However, there may be an *excess capability penalty (p_{capBp})* on system B in production and operations because the components that are reused may be overdesigned for system B, depending on which system is more capable.

The production portion of the system is composed of labor costs and materials cost. The labor portion of production is benefitted from *learning* (L_{AB}), while the materials portion is benefitted by *economies of scale (eos).*

Learning benefits are related to the learning that occurs through repetition of the same tasks, resulting in decreased time or cost to conduct the task. Learning can be incorporated into the model using the learning curve equation (Equation 1) (Newman, Eschenback, & Lavelle, 2004), in which T_N is the labor for the Nth unit compared to the labor for the initial unit, T_i. The fractional learning curve rate is indicated in the equation as *learning*, which has a typical value between 0.75 and 0.95 (NASA Cost Estimating Handbook, 2008).

Equation 1: Learning Curve

$$T_N = T_i * L$$

$$L_A = N_A^{\frac{\log(learning)}{\log(2.0)}}$$

$$L_{AB} = (N_A + N_B)^{\frac{\log(learning)}{\log(2.0)}}$$

If system A and system B are produced simultaneously and use the same component, both systems derive more benefit and share equally in the benefits from learning. If there is offset, the system that is developed later obtains greater benefits from learning than the first system because production of the common components in the later system begins farther down the learning curve. Based on the simple learning curve equation with a 0.85 learning curve, Figure 36 shows the relative benefits of learning in parallel, with a 50% offset, and sequential production. The yellow and orange graphs in Figure 36 indicate the unit cost with learning benefits from systems developed independently, with no benefits from common learning. The red and blue graphs indicate the unit cost with accelerated learning benefits for systems developed in common, i.e. with accelerated common learning benefits.

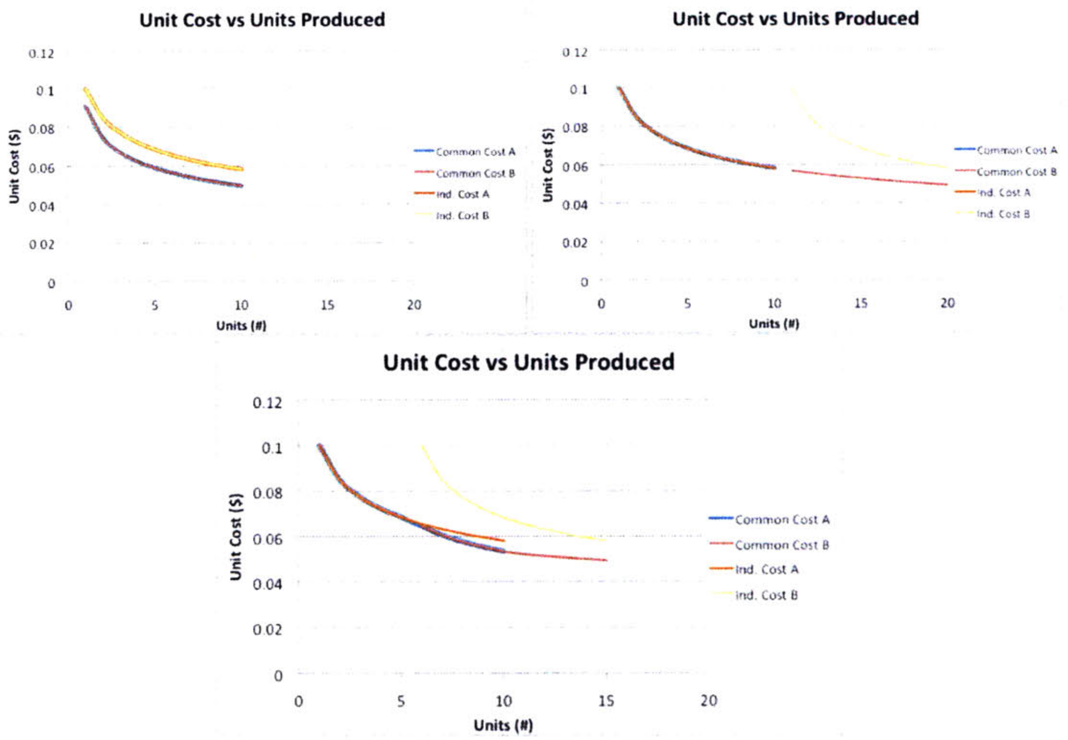


Figure 36: Relative learning benefits on the unit cost between common development and independent development for production in parallel (top left), 50% offset (bottom center), and sequentially (top right)

The *Economies of scale* factor represents the benefit obtained from purchasing or producing more components in the same amount of time. For example, it would cost less per unit to produce or procure 20 units compared to 5 units of a component in a given year. *Economies of scale* are incorporated into the economic model using Equation 2, in which N represents the years production, N_{total} is the total production for the entire life cycle, T_i represents the average unit cost without economies of scale, and T_N represents the average unit cost with economies of scale. Based on the economies of scale equation, Figure 37 presents the benefits of economies of scale based on the production rate and Figure 38 demonstrates how economies of scale increases the benefits of commonality in years in which production overlaps.

While Equation 2 will suffice for this example, the method of incorporating economies of scale should be more precisely defined by the particular system's production function (de Neufville, 1990).

Equation 2: Economies of Scale

$$T_N = T_i * eos$$

$$eos = 0.7 * e^{\frac{-2*N}{N_{total}}} + 0.3$$

$$eos_{AB} = 0.7 * e^{\frac{-2*(N_A + N_B)}{(N_{totA} + N_{totB})}} + 0.3$$

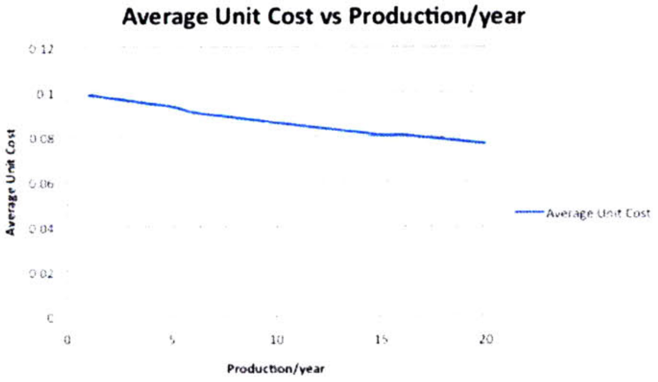


Figure 37: Average unit cost vs the production rate with economies of scale benefits

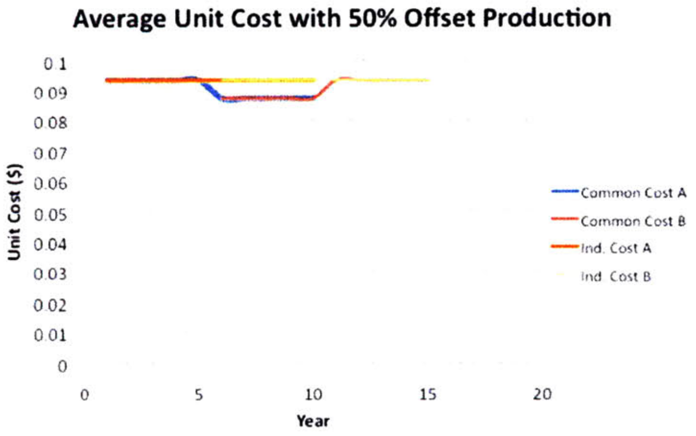


Figure 38: Economies of scale benefits in which production is 50% offset

Operations

The operations portion of the system is composed of fixed recurring and variable recurring costs. Operations fixed recurring costs often include sustaining engineering and operations infrastructure, while variable recurring costs often include operations labor, training, and logistics. The variable recurring costs in operations are benefitted by accelerated *learning* (L_{AB}), similar to the production costs described above. The fixed recurring costs are benefitted by a *capability overlap* (cap_o). The capability overlap is the benefit created by an overlap in fixed recurring costs. For the same components used by multiple systems the fixed recurring costs of the components are not duplicated because they do not depend on the number of operations per year, but only on the ability to operate the component. The economic model assumes that system A and B split the benefits from the capability overlap based on the operation rate of each system. The benefits of the capability overlap could have been assigned completely to system B, but doing so may artificially penalize system A and benefit system B. The distribution of benefits does not affect the total life cycle costs.

Table 10: Factors that affect the cost of class 3 components

Class		**Development**		**Production**		**Operations**	
#	**Desc.**	**Sys. A**	**Sys. B**	**Sys. A**	**Sys. B**	**Sys. A**	**Sys. B**
3	IUA ➔C	1	p_{int}	$(L_{AB})_A$, eos	p_{capBp}, $(L_{AB})_B$, eos	$(L_{AB})_A$, cap_o	p_{capBo}, $(L_{AB})_B$, cap_o

Class 4: Intended Common to Common

Development

This class of components is developed with the intent that the components become common between the systems and then they actually become common. This class has an upfront *cost of common development* (p_{cd}) on system A, as in class 2, and an *integration penalty* (p_{int}) on system B, as in class 3.

Production and Operations

The production and operation phases of this class are marked by the possibility of an *excess capability penalties* (p_{cap}) on both system A and system B. In addition, the labor portion of the production phase on both systems benefit from *learning* (L_{AB}) and the materials portion in the production phase of both systems benefit from *economies of scale (eos)*. The variable recurring costs of the operations phase are benefitted from learning (L_{AB}), while the fixed recurring costs are reduced by a *capability overlap* (cap_o) as in class 10.

Table 11: Factors that affect the cost basis for class 4 components

Class		Development		Production		Operations	
#	**Desc.**	**Sys. A**	**Sys. B**	**Sys. A**	**Sys. B**	**Sys. A**	**Sys. B**
4	IC →C	p_{cd}	p_{int}	p_{capAp}, $(L_{AB})_A$, eos	p_{capBp}, $(L_{AB})_B$, eos	p_{capAo}, $(L_{AB})_A$, cap_o	p_{capBo}, $(L_{AB})_B$, cap_o

Class 5: Unique B

At the time that system B is developed components that are developed to be unique to system B are unique. As a result, the independent cost estimate for this class passes through the model without any modification.

Table 12: Factors that affect the cost basis for class 5 components, in which a '0' indicates that the basis is not part of the cost estimate and a '1' indicates that the basis is not affected by the benefits and penalties of commonality

Class		Development		Production		Operations	
#	**Desc.**	**Sys. A**	**Sys. B**	**Sys. A**	**Sys. B**	**Sys. A**	**Sys. B**
5	UB	0	1	0	1	0	1

Summary

Table 13 summarizes each of the benefits and penalties of commonality that are included in the economic cost model. In the table green indicates a probable benefit and red indicates a probable penalty to the system. However, Table 13 does not show the specific formulas used to create the model, or all factors considered in the model, but only the factors that are in place to analyze the relative benefits of commonality. In addition to these factors, economies of scale and learning benefits for independently developed components were also incorporated into the model.

Table 13: Summary of each of the effects of commonality on the system based on the development phase

Class		Development		Production		Operations	
#	Desc.	Sys. A	Sys. B	Sys. A	Sys. B	Sys. A	Sys. B
1	IUA→UA	1	0	1	0	1	0
2	IC →UA	p_{cd}	0	p_{capAp}	0	p_{capAo}	0
3	IUA →C	1	p_{int}	$(L_{AB})_A$, eos	p_{capBp}, $(L_{AB})_B$, eos	$(L_{AB})_A$, cap_o	p_{capBo}, $(L_{AB})_B$, cap_o
4	IC →C	p_{cd}	p_{int}	p_{capAp}, $(L_{AB})_A$, eos	p_{capBp}, $(L_{AB})_B$, eos	p_{capAo}, $(L_{AB})_A$, cap_o	p_{capBo}, $(L_{AB})_B$, cap_o
5	UB	0	1	0	1	0	1

By considering each of the benefits and penalties integrated into the model the cost impact that each class will have on each phase of the system can be determined. Figure 39 and Figure 40 capture the relative cost impacts compared to independent development of each class, based on each of the effects described in the above section.

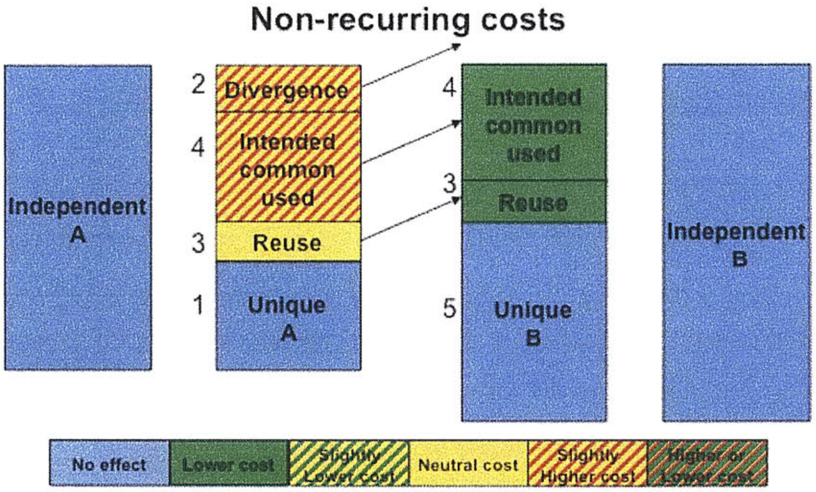


Figure 39: The relative cost impacts of commonality on each class of components on the non-recurring costs as a result of the described benefits and penalties

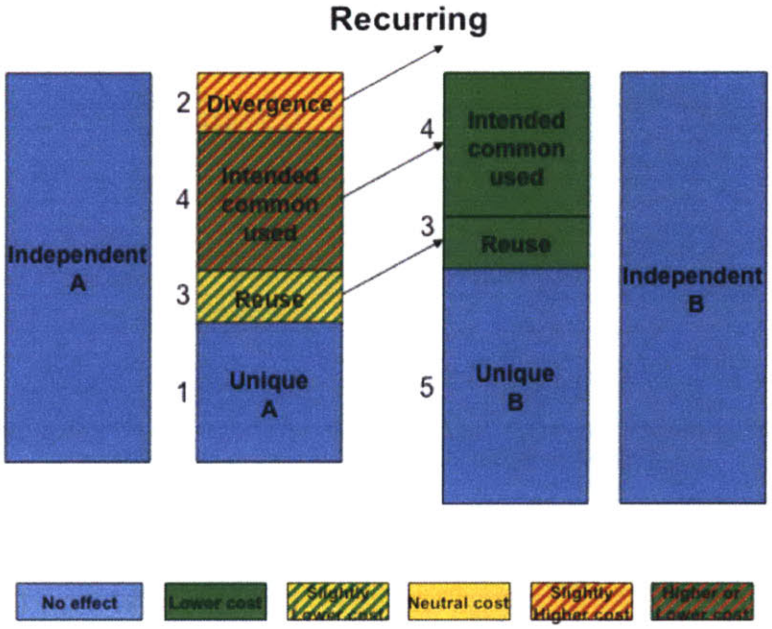


Figure 40: The relative cost impacts of commonality on each class of components on recurring costs as a result of the described benefits and penalties

All of the benefits and penalties of commonality are incorporated into the economic model by applying the factors described above to alter the original cost basis. Table 14, Table 15, and Table 16 show the specific formulas used for the cost basis, and the multipliers for the development, production, and operations phases of the system. The factors in the formulas that are not defined in the above section are described in Table 17.

Table 14: Equations used to analyze opportunities for commonality, including the cost basis and the system multipliers

			Development Multiplier	
Class	**Desc.**	**Cost Basis**	**System A**	**System B**
1	IUA→UA	$A*(1-C)*(1-r_{iu})$	1	0
2	IC →UA	$A*C*d_{iv}$	$(1+p_{cd})$	0
3	IUA →C	$A*(1-C)*r_{iu}$	1	p_{int}
4	IC →C	$A*C*(1-d_{iv})$	$(1+p_{cd})$	p_{int}
5	UB	$B-A*(C*(1-d_{iv})-(1-C)*r_{iu})$	0	1

Table 15: Multiplier equations used in the economic model to incorporate the benefits and penalties of commonality for the production portion of the system

		Production Multiplier	
Class	**Desc.**	**System A**	**System B**
1	IUA→UA	$N_A*r_p*(r_{mat}*eos_A+(1-r_{mat})*L_A)$	0
2	IC →UA	$N_A*r_p*(1+p_{capAp})*(r_{mat}*eos_A+(1-r_{mat})*L_A)$	0
3	IUA →C	$N_A*r_p*(r_{mat}*eos_{AB}+(1-r_{mat})*(L_{AB})_A)$	$N_B*r_p*(1+p_{capBp})*(r_{mat}*eos_{AB}+(1-r_{mat})*(L_{AB})_B)$
4	IC →C	$N_A*r_p*(1+p_{capAp})*(r_{mat}*eos_{AB}+(1-r_{mat})*(L_{AB})_A)$	$N_B*r_p*(1+p_{capBp})*(r_{mat}*eos_{AB}+(1-r_{mat})*(L_{AB})_B)$
5	UB	0	$N_B*r_p*(r_{mat}*eos_B+(1-r_{mat})*L_B)$

Table 16: Multiplier equations used in the economic model to incorporate the benefits and penalties of commonality for the operations portion of the system

		Operations Multiplier	
Class	**Desc.**	**System A**	**System B**
1	IUA→UA	$r_f+N_{OA}*r_v*L_A$	0
2	IC →UA	$(r_f+N_{OA}*r_v*L_A)*(1+p_{capAo})$	0
3	IUA →C	$r_f*cap_o+N_{OA}*r_v*(L_{AB})_A$	$(r_f*cap_o+N_{OB}*r_v*(L_{AB})_B)*(1+p_{capBo})$
4	IC →C	$(r_f*cap_o+N_{OA}*r_v*(L_{AB})_A)*(1+p_{capAo})$	$(r_f*cap_o+N_{OB}*r_v*(L_{AB})_B)*(1+p_{capBo})$
5	UB	0	$r_f+N_{OB}*r_v*L_B$

Table 17: Previously undefined factors required for the quantitative analysis of the systems

N_A	Units of production in System A
N_B	Units of production in System B
r_p	Ratio of first unit of production cost to development cost
r_{mat}	Fraction of the system cost related to materials
r_f	Ratio of fixed recurring operations cost to development cost
r_v	Ratio of variable recurring operations cost to development cost
N_{OA}	Number of operations of System A
N_{OB}	Number of operations of System B

7.2.2 Cost Model Application

In order to better demonstrate the effects of commonality that are incorporated into the economic model, the model is applied to the Constellation Space Suit System (CSSS). Specific cost information on the CSSS is not publicly available, as a result some assumptions must be made in order to quantitatively analyze the CSSS. Experienced managers with intimate knowledge of the system would be able to better quantify the assumptions in the model. The analysis is conducted through a three-step process: (1) characterize the system, (2) execute the cost model, and (3) analyze the results.

Step 1: Characterize the System

The first step in analyzing the value of the common system is to characterize the system, by determining the life cycle timeline, the independently developed cost estimates for each phase of the life cycle, and estimates for the model inputs that relate to the impact of commonality on the system.

The first system variant in the CSSS is the configuration 1 suit, expected to cost $180 million to develop. The second system is the configuration 2 suit, which will be assumed to cost $350 million to develop independently (NASA, 2008). Production and operations costs will then be assumed to be a function of the systems development cost. The development costs for both systems are distributed evenly across the development time span.

The first unit of production will be assumed to cost 10% that of development. 85% of the production cost will be associated with labor while 15% is associated with materials. Production is assumed to be conducted for five years for each system with a production rate of 7 per year for the configuration 1 suit and 6 per year for the configuration 2 suit.

The operations fixed recurring costs will be assumed to be 5% of development per year and variable recurring costs to be 3% of development per mission. Each mission to LEO will require configuration 1 suits, while each mission to the moon will require both configuration 1 and configuration 2 suits. It is assumed that there are 2 missions to LEO and 2 missions to the moon per year during operations. The timeline for development, production, and operations is shown in Figure 41, based on the initial ESAS report (ESAS, 2005). Operations of the configuration 1 suit will run from 2015 to 2025 and operations of the configuration 2 suit will begin by 2020 and continue through 2025. Table 18 summarizes the factors applicable to the life cycle's development, production and operations timeline.

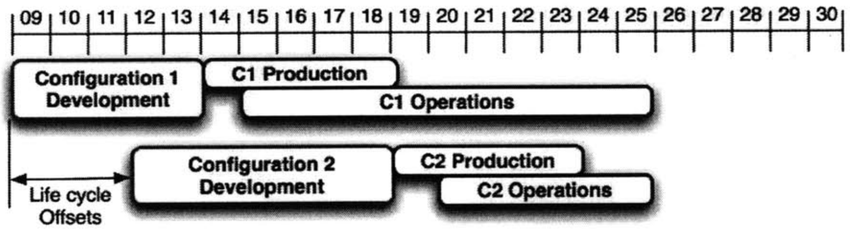


Figure 41: Assumed life cycle timeline for the CSSS example

Table 18: Summary of the system's development, production, and operations timeline for the CSSS example

Life Cycle Phase	Start Date	Duration (yrs)	Rate	Units
Development A	2009	5	20%	%/year
Production A	2014	5	7	Quantity/year
Operations A	2015	10	2	Ops/year
Development B	2012	7	14.2%	%/year
Production B	2019	5	6	Quantity/year
Operations B	2020	5	2	Ops/year

In order to characterize the effects of commonality on the system it is first important to evaluate how much of system A is being developed with the intent to become common. The initial goal for the CSSS was to reuse as much of the system as possible, including components, technologies, and processes. Without specific cost information for particular aspects of the system, it will be assumed that 60% of system A will be developed with the intent that it becomes common. The other assumptions that were made for the analysis are summarized in Table 19.

Table 19: Summary of each of the assumed factors in the analysis

Development Cost Factors	**Factor**	**Quantity**
Development Scope A	A	180
Development Scope B	B	350
Intended Commonality	C	0.6
Common Development Penalty	p_{dev}	0.1
Integration Penalty	p_{int}	0.1
Reuse of Intended Unique	r_{iu}	0.1
Divergence	d_{iv}	0.15

Production Cost Factors	**Factor**	**Quantity**
Ratio (mfg. unit cost/dev cost A)	r_p	0.1
Ratio (mfg. unit cost/dev cost B)	r_{pB}	0.1
Ratio (material/total costs)	r_{mat}	0.15
Learning Curve	plearning	0.85
Economies of scale	eos	0.7
Excess Capability Penalty A	p_{capA}	0.15
Excess Capability Penalty B	p_{capB}	0.1

Operations Cost Factors	**Factor**	**Quantity**
Ratio (ops. Fixed/Dev) A	r_{fA}	0.05
Ratio (ops. Variable/Dev) A	r_{vA}	0.03
Ratio (ops. Fixed/Dev) B	r_{fB}	0.05
Ratio (ops. Variable/Dev) B	r_{vB}	0.03
Operations Learning	olearning	0.95
Capability Overlap	cap_o	1

Step 2: Execute the Cost Model

The cost model shows that the common system would have an undiscounted life cycle cost through 2025 of $1,850 million while the independently developed system will cost $1,904 million, as seen in Table 20. The annual costs for each phase in the life cycle can be seen in Figure 42, for both independent and common development.

Table 20: Results from the economic cost model, prices in $ millions

Development Type	System	Development	Production	Operations	Sys. Total	Total
Common	A	$190.80	$405.87	$212.31	$808.98	$1,849.88
	B	$260.90	$613.49	$166.51	$1,040.90	
Independent	A	$180.00	$372.36	$211.11	$763.47	$1,903.70
	B	$333.80	$635.31	$171.12	$1,140.23	

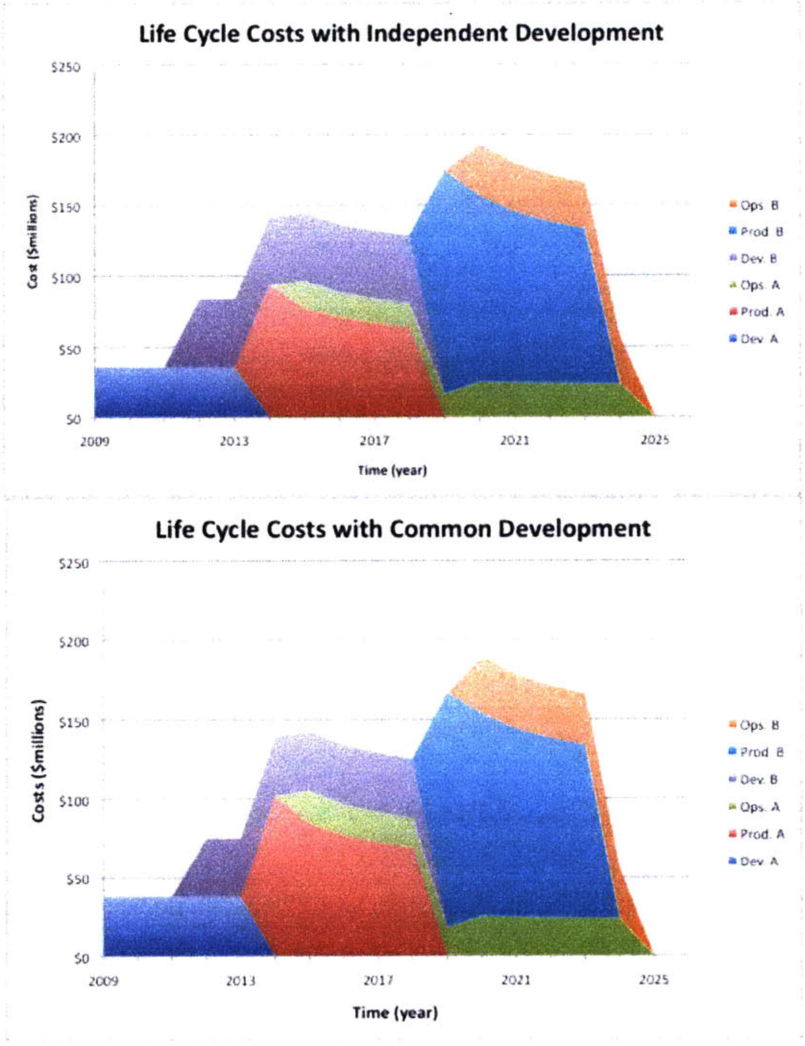


Figure 42: Sand charts of the development, production, and operations costs with independent development (top) and common development (bottom)

Step 3: Analyze the Results

The results from the cost model show that compared to independently developed systems the life cycle cost of a common system B and the total life cycle cost of both common systems are lower, while the life cycle cost of a common system A is higher. However, the higher life cycle cost of the common system A is misleading because the entire penalty is not paid up-front. In fact, the only years in which costs with common development are higher than costs with independent development are the first three years, before the development of system B begins, as seen in Figure 43. As a result, the up-front penalty is actually $6.4 million, much less than Table 20 implies. In addition, the entire up-front penalty is regained in year 4. Based on the $6.4 million dollar investment, the Return on Investment (ROI) for the common system is **8.4** with a 4-year pay back period.

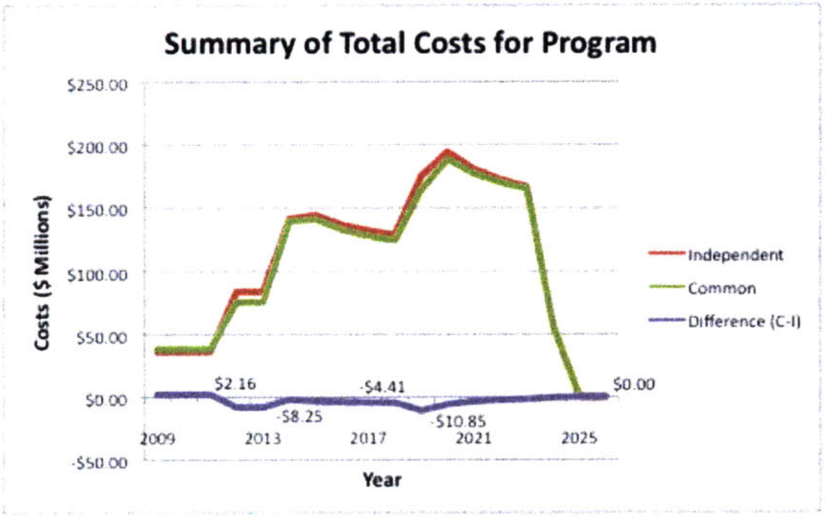


Figure 43: Life cycle cost for independent development, common development, and the difference between the two development estimates

Affect of the Benefits and Penalties

When developing a system with commonality it is important to understand how each of the factors considered in the analysis, affect the life cycle costs of the system. To demonstrate the relative impact of each factor, all factors were removed from the model and then re-added, while the change to the life cycle cost was recorded. Table 21 summarizes each factor that was added and the cost impacts on the life cycle cost estimates for independently developed systems and systems developed with commonality.

Table 21: Summary of the how each benefit and penalty factor considered in the model affects the resultant total life cycle cost estimate

		Common	**Independ.**	**Delta Comm.**	**Delta Ind.**	**Ind-Comm Difference**
Benefit/Penalty	**Factor**	$2,641.68	$2,641.68			$0.00
Commonality	0.6	$2,541.37	$2,641.68	-$100.31	$0.00	$100.31
Divergence	0.15	$2,557.57	$2,641.68	$16.20	$0.00	$84.11
Reuse	0.1	$2,550.89	$2,624.96	-$6.69	-$16.72	$74.08
Cost of CD	0.1	$2,561.69	$2,624.96	$10.80	$0.00	$63.28
Integration Penalty	0.1	$2,571.59	$2,626.76	$9.90	$1.80	$55.18
Excess Capability A	0.15	$2,642.06	$2,626.76	$70.47	$0.00	-$15.29
Excess Capability B	0.1	$2,677.20	$2,632.16	$35.15	$5.40	-$45.04
Production Learning	0.85	$1,963.24	$1,985.93	-$713.96	-$646.23	$22.69
Economies Of Scale	0.7	$1,934.77	$1,965.97	-$28.47	-$19.96	$31.19
Capability Overlap	100%	$1,918.05	$1,965.97	-$16.72	$0.00	$47.92
Operations Learning	0.95	$1,849.88	$1,903.70	-$68.17	-$62.27	$53.82

Affect of the Discount Rate

For the presented economic cost evaluation a discount rate of 0% was used. A discount rate of 0% is often used on government programs and public goods that cannot be completely evaluated with monetary values. However, it can also be argued that the government should use a discount rate for economic evaluation, at either a rate comparable to the interest rate of bonds (around 5%) or at the rate used by industry because the government obtains money from industry in the form of taxes (around 10%) (de Neufville, Engineer Systems Analysis for Design). Therefore, the model was rerun to assess the affect of the discount rate. The resultant expected costs with varying levels of discount are shown in Table 22.

The results show that while the discount rate does decrease the benefits of commonality slightly, there is no change in the preferred development method. The discount rate has relatively little effect on the results, because common development offers consistent benefits throughout the life cycle. However, if the offset were larger than the discount rate may have more effect on the results.

Table 22: The expected values of the system with varying levels of discount

Discount Rate	Development Type	System A	System B	Discounted Life Cycle Costs
0%	Independent	$763	$1,140	$1,904
	Common	$809	$1,041	$1,850
5%	Independent	$557	$699	$1,256
	Common	$591	$629	$1,221
10%	Independent	$424	$450	$874
	Common	$451	$399	$850

Limitations

Several assumptions must be made in order to conduct the economic evaluation. As a result, little is known about the certainty of the estimated life cycle cost for both types of development. Additionally, because this method of estimating costs begins with the independent cost estimates, the common system cost estimates are only as good as the independent cost estimate inputs to the model.

7.3 Options-based Economic Model

While the deterministic cost model analysis does present useful information in regards to the expected life cycle cost, it does not present any information related to the certainty of the expected outcome or development flexibility. In order to improve the economic analysis and specifically consider flexibility and options in the system, an options-based model was developed. Options or flexibility in the system are valuable when there is uncertainty, because the option allows developers to take advantage of beneficial events of uncertainty, while avoiding unbeneficial events (Neely & de Neufville, 2001).

This section will present the process used to evaluate the economic impacts of commonality in the system, as a Real Option. The analysis will be discussed as it is applied to the CSSS example, also presented in the previous section. To conduct the analysis, four additional steps are required: (4) identify the uncertainties and options in the system, (5) determine the analysis method, (6) quantify the uncertainties and options, and (7) execute the analysis.

Step 4: Identify Uncertainties and Options

The first step in conducting the economic analysis is to identify the uncertainties and options that exist in the system. There are several uncertainties related to the evaluation, so instead of attempting to identify all of the uncertainties this analysis will focus on the greatest uncertainties or the uncertainties that are believed to have the greatest affect on the system. The farther in the future an event is to occur, the more uncertain the event is. Therefore, the most uncertain events are related to system B. For the CSSS three uncertainties were evaluated:

- *Start of development for System B* – The start of development for system B controls the size of the development offset. Increasing the offset decreases the benefits to production and operations phases and increases the likelihood that divergence will occur. There are several factors that could delay development, including a change in architecture, similar to the one that is currently being considered in response to the report issued by the Review of U.S. Human Space Flight Plans Committee (Review of U.S. Human Spaceflight Plans Committee , 2009).
- *Divergence* – The second uncertainty involves the amount of divergence that will occur. Divergence has been observed in all three NASA case studies and seven industry case studies, however it is believed that the size of the offset between systems contributes to the amount of divergence that will occur.
- *Scope of System B* – The final uncertainty evaluated in this model is the scope of development for system B. Large government projects have become infamous for cost overruns. Therefore, one of the greatest uncertainties that can influence total life cycle cost is the scope of system B. In a fixed budget project this translates into an extension in development time for the system.

In addition to these uncertainties, the system also contains options that offer additional flexibility to management. Offsets in the system create a decision at the time that the first system is developed, to either (1) *invest in commonality* or (2) *develop the system independently*. By investing in commonality at the time that the first system begins development, developers have created the option at the time that the second system is being developed to either (1) *cut their losses and develop independently,* or (2) *continue to invest in and develop commonality.* The option to cut losses and develop independently becomes valuable if there is a large change in the system that increases the likelihood that divergence will occur.

Step 5: Determine the Analysis Method

Three different analysis methods were explored to evaluate the systems: Net Present Value (NPV), Decision Analysis (DA), and Real Options Analysis (ROA) methods. There are two factors to consider when determining the analysis method: (1) the objective of the analysis and (2) the type of uncertainty in the system that was identified in step 4 (Nilchiani & Hastings, 2007).

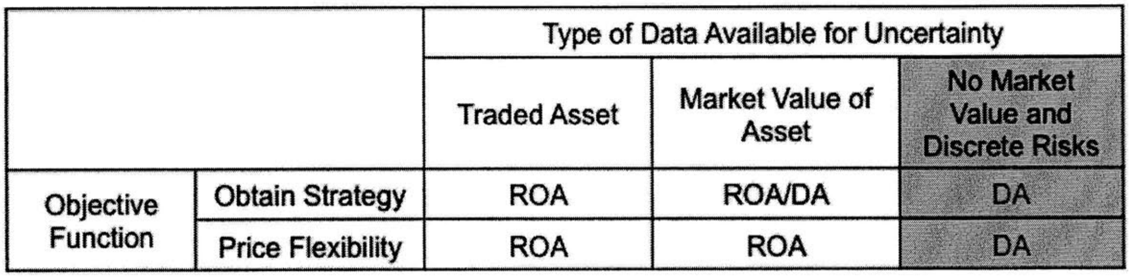
The first consideration when choosing between the valuation methods is the objective. The primary objectives of the current evaluation are (1) to price the value of the option and (2) to create a strategic development plan. Both, the decision analysis and the real options analysis methods can be used to value the option, however only the decision analysis method will allow us to create a strategic development plan, as shown in Table 23.

Table 23: Objective of analysis compared with the different valuation methods (de Neufville)

		Valuation Methods		
		Net Present Value	Decision Analysis	Real Options Analysis
Objective Function	Point Estimate of Project Value	Yes	Yes	No
	Obtain Strategy	No	Yes	No
	Price Flexibility	No	Yes	Yes

The other factor to consider when evaluating the various options is the type of uncertainty in the system, demonstrated by Table 24. If the uncertainty in the system can be priced as a traded asset, such as the price of oil or another consumable, it may be more appropriate to use real options analysis. However, if the uncertainty is discrete and no market price is available than decision analysis is the more appropriate analysis method. This evaluation will be conducted using the decision analysis method. If the system being evaluated contains a combination of uncertainty types than a hybrid method can be used (Neely & de Neufville, 2001).

Table 24: Comparison of the type of uncertainty in the system to the objective of the analysis (de Neufville)



		Type of Data Available for Uncertainty		
		Traded Asset	Market Value of Asset	No Market Value and Discrete Risks
Objective Function	Obtain Strategy	ROA	ROA/DA	DA
	Price Flexibility	ROA	ROA	DA

Step 6: Quantify the Uncertainties and Options

In order to conduct the decision analysis a decision tree was created that includes quantified values for each of the uncertainties and options identified in Step 4. Figure 44 shows the decision tree that was developed, in which the blue boxes represent decisions, the green ovals represent an uncertain event, and the red boxes represent discrete outcomes.

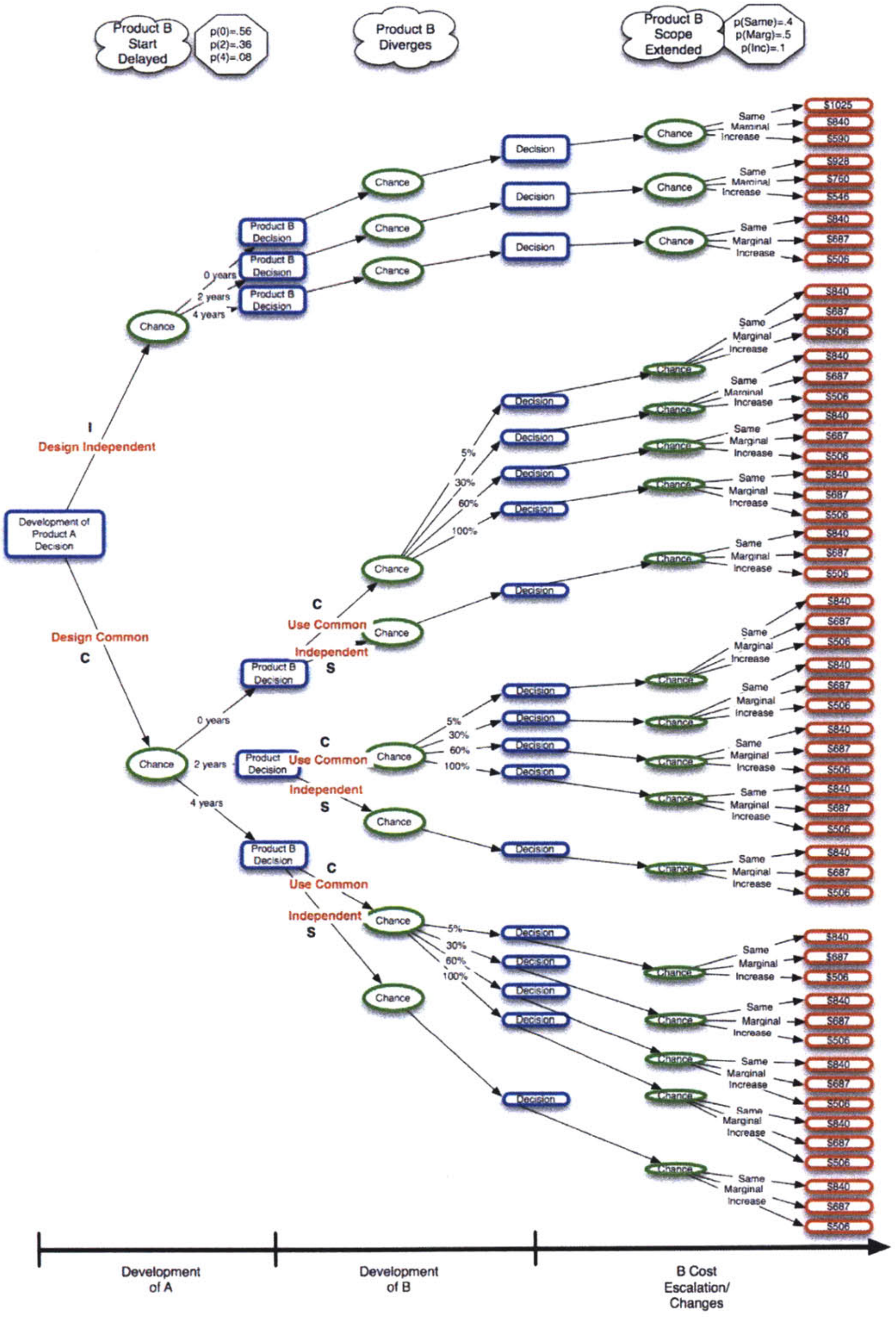


Figure 44: Decision tree developed for the CSSS options-based analysis

In the decision tree, the first decision is to either develop system A independently or with intended forward commonality. The uncertain factor that follows is the size of the offset between system A and system B. The second decision once the development of system B begins is to either utilize the previously developed common components for system B, or to stop common development in system A and begin developing both systems independently. The following uncertainty is the amount of divergence that occurs. There is no decision in the final set of blue boxes, however an additional decision could easily be integrated into the analysis if needed to better frame the system. The final uncertainty is the scope or extension for the development of system B. The red boxes then represent each discrete outcome that could result.

In order to quantify the decision tree analysis, the probability of each discrete event occurring should be estimated. The assumed probability for each uncertainty is listed in

Table 25. Divergence was chosen to be a function of the offset. Therefore, the probabilities chosen for the analysis are represented in Figure 45, showing that as the offset increases the probability that a larger amount of divergence will occur increases.

Table 25: Assumed probabilities of discrete events used for the CSSS analysis

	Outcome	Probability
Delay of Start of System B Development	0 years	0.5
	2 years	0.4
	4 years	0.1
Divergence (offset = 0 years)	5%	0.4
	15%	0.35
	60%	0.2
	100%	0.05
Extension of System B Development	0 years	0.4
	2 years	0.4
	4 years	0.2

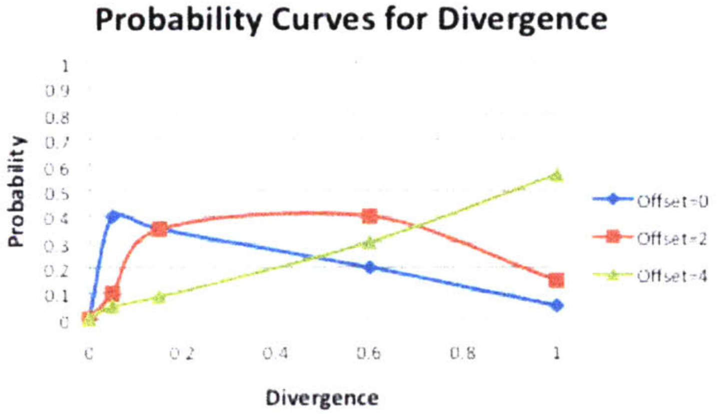


Figure 45: Description of probabilities chosen for the discrete decision tree

Step 7: Execute the Analysis

Each of the discrete events outlined in the decision tree has an associated probability with that outcome occurring. Therefore the decision tree can be used to formulate a Cumulative Distribution Function (CDF) for each of the choices: *independent development*, *common development*, and initial common development *switched to independent* development, shown in Figure 46. Each of the CDFs correlates to branches in the decision tree, indicated by either a 'I', a 'C', or a 'S', in Figure 44. The CDF allows management to determine the probability that the life cycle cost will be at or below a certain value. The curves show that while the common system has the lowest expected life cycle cost (Table 26), developing with commonality also presents the lowest and highest possible life cycle costs.

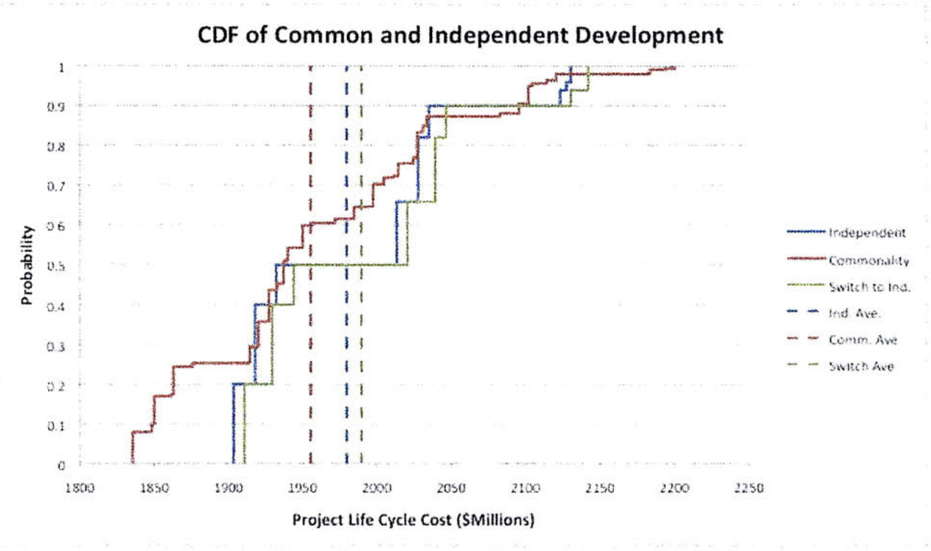


Figure 46: CDFs for each of the choices and the expected costs of each option

Table 26: Values for the expected cost of each CDF curve

Development Method	Expected Cost (EC)
Independent	$1,979.79
Commonality	$1,955.58
Switch to Ind.	$1,989.69
Best Approach	$1,952.70

However, the described CDFs do not present the best approach for development. In order to determine the best approach for development, the decision tree in Figure 44 should be analyzed by "rolling back" from the discrete outcomes to quantify the expected value of each branch. Therefore, by implementing the decision rule to choose the branch that has the lowest expected cost, the best development approach can be determined. The analysis will not only determine the best approach for the development of commonality, but also determine the value of the flexibility that the option to utilize the developed commonality or choose not to utilize the commonality introduces to a system.

The best development strategy is to begin development of system A with intended commonality. However, as additional information is obtained on the start of development for system B and the size of the offset, the development of commonality within system A is either continued or cancelled. The decision tree shows that the best strategy is to develop with commonality and if the start of development for system B is extended by four or more years to discontinue effort on common development, as shown in Figure 47.

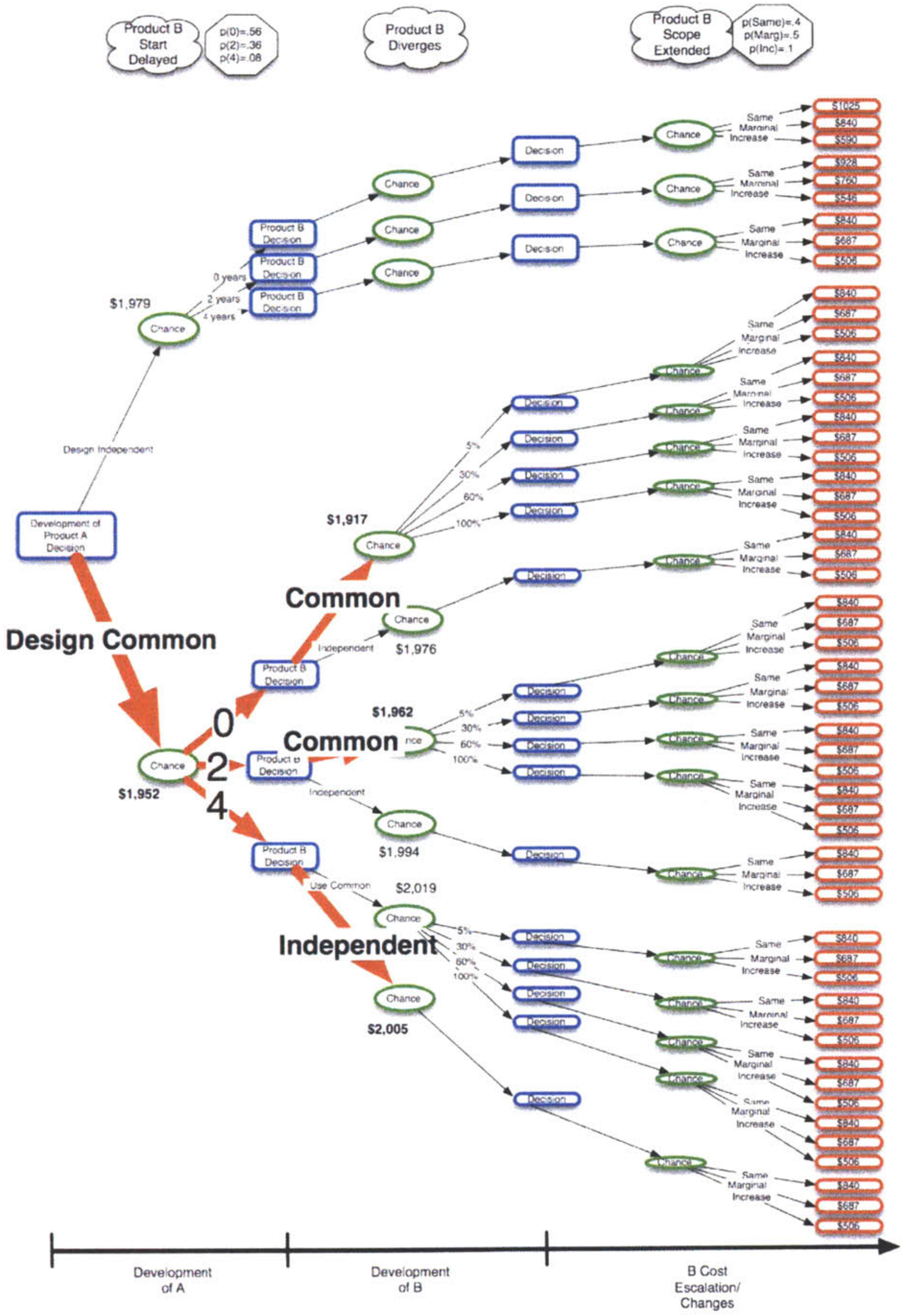


Figure 47: Development strategy based on decision tree analysis

If the option to switch to independent development is practiced than a new CDF results, identified as the best approach in Figure 48. The best approach shifts portions of the common development CDF to the left, lowering the expected life cycle cost by nearly $3 million. Therefore, the option to switch to independent development offers an additional value of $3 million to the system, as shown in Table 27. The lower expected life cycle costs show that considering flexibility and the Real Option in the system can add value and affect the optimal development approach.

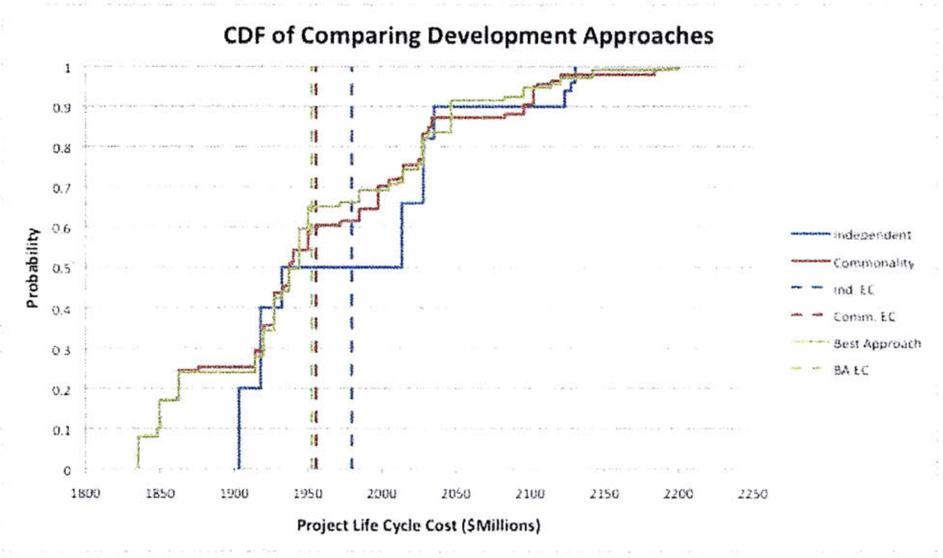


Figure 48: Chart comparing previous development CDFs to the "Best Approach" CDF

Table 27: Comparison of the expected cost of each development approach

Development Method	Expected Cost (EC)
Independent	$1,979.79
Commonality	$1,955.58
Switch to Ind.	$1,989.69
Best Approach	$1,952.70

7.4 Conclusion

Commonality is not an end in itself, but a means to obtain benefits. Commonality offers both benefits and penalties to the entire life cycle of a system that should be considered before a decision is made to either implement or not implement an opportunity for commonality. However, the three NASA case studies and six of the seven industry case studies conducted (Boas, 2008), showed that current evaluation methods do not effectively consider the specific benefits of commonality. This chapter presented two economic evaluation methods that update independent cost estimates by integrating the specific benefits and penalties of commonality to create cost estimates for common systems.

In this chapter, both a deterministic cost model and a cost model to evaluate commonality as a Real Option are presented. The cost models are also demonstrated by applying each method to the CSSS case study. While the deterministic economic model does reveal information about the up-front penalty and relative benefits of commonality throughout the life cycle, it does not present any information on the certainty of life cycle costs or the options available to management.

Evaluating commonality as a real option builds on the deterministic method by additionally considering uncertainties and flexibility in the system. Commonality introduces both risk and uncertainty to a system that can offer additional value. By evaluating commonality as a real option and considering the additional flexibility the expected life cycle costs are reduced by nearly $3 million.

The decision tree analysis conducted in the chapter enables the development of a strategic development plan for the common system. The decision tree analysis also showed which uncertain events had the greatest impact on the development strategy. While possible increases in the development cost of system B do not affect the development strategy, the size of the offset and the probability that divergence will occur greatly impact the development strategy.

There are several limitations to the described evaluation. In order to quantify the analysis, several assumptions were required. System managers that have more information on the system would be better equipped to estimate the described assumptions. In addition, a limited number of generic uncertainties and assumptions were explored, but there may be additional system specific uncertainties that could have a larger impact on development.

While there are limitations to the analysis considering the uncertainties and options in the system is far better than ignoring them. Considering options gives management useful insight into the affects of commonality, the possible outcomes of development, and how to best manage the possible outcomes. For example, it may be determined that by investing more money in the intended common development of system A and paying a higher *cost of common development,* the *integration penalty* and probability that divergence will occur may decrease; adding value to the system.

8. Conclusion

Commonality has the potential to greatly benefit systems, but has proven difficult to manage. The goal of this thesis was to determine how commonality should be managed within NASA systems and what the differences are between NASA and industry applications of commonality. This chapter will summarize the work presented in the thesis, present conclusions and findings based on of the presented work, and outline suggestions for future research on the application and management of commonality.

8.1 Thesis Summary

In order to determine how commonality should be managed within NASA systems, it is first necessary to determine how commonality is currently being managed within NASA systems. To that end, three case studies of past and present NASA systems were conducted on (1) the Constellation Space Suit System (CSSS), (2) the International Standard Payload Rack (ISPR), and (3) the Core Flight-Software System (CFS). Each of the case studies is documented as an individual chapter in the thesis with a detailed description of the system, challenges to the management of commonality, and observed methods to manage commonality.

After the three NASA case studies were completed and the individual case study analysis was conducted, cross-case analysis was performed between the three NASA case studies and the seven industry case studies. The cross-case analysis was performed in order to determine the similarities and differences between the trends in commonality, key challenges, and management methods observed in the NASA and industry case studies. The cross-case analysis revealed a set of managerial guidance principles to assist with future applications of commonality. The guidance, described in detail in Chapter 6, discusses the management of commonality in regards to the three critical management tasks: (1) management of the identification of opportunities, (2) the evaluation of opportunities, and (3) the implementation of beneficial opportunities. The guidance also addresses several system characteristics, including the system type, organizational structure, development method, and the relative capability and production volume of the system variants.

The case studies revealed a weakness in the current evaluation process to determine whether or not an opportunity for commonality is beneficial. To address the weakness two economic cost models were developed, a deterministic economic cost model and an economic cost model that evaluates commonality as a Real Option. The economic models transform cost estimates for independently developed systems into cost estimates for common systems by specifically accounting for the economic benefits and penalties of commonality. Evaluating commonality as a Real Option, additionally takes into account uncertainties and flexibility in the system to create a strategic development plan. Both the deterministic and Real Options economic cost models are applied to an example of the CSSS case study.

8.2 Key Findings

The three NASA case studies, the cross-case analysis between the three NASA and seven industry case studies, and the development of the economic cost models have yielded a number of findings and conclusions with regards to the management of commonality. The findings based on the work presented in this thesis are discussed below.

- Commonality will not be created in a system by chance, but instead must be actively sought out.
- Commonality is not always beneficial, but must be evaluated to determine the benefits. Commonality is a method of trading between competing goals, often by obtaining cost and risk benefits while sacrificing system performance. As a result, opportunities for commonality must be evaluated to determine if there is a net benefit.
- Life cycle offsets and divergence were observed in all three NASA case studies and seven industry case studies (Boas, 2008). The factors that contribute to the existence of life cycle offsets and divergence were found to be common between

all NASA and industrial case studies, and likely universal to all applications of commonality. As a result, offsets and divergence can be expected in all applications of commonality.

- Three management approaches were observed in the three NASA and seven industry case studies:
    1. Reactive reuse
    2. Common building block approach
    3. Widespread application of forward commonality

- In order for commonality to be applied successfully into a system, three managerial tasks are required:
    1. Management of the identification of opportunities
    2. Evaluation of opportunities
    3. Implementation of opportunities that are deemed beneficial

- Specific managerial guidance has been developed to assist with the application of commonality into future systems:

    Guidance 1: The identification process must begin early in development in order to maximize the potential benefits.

    Guidance 2: Design with flexibility to reduce the effect of uncertainty on the system.

    Guidance 3: Reduce uncertainty in future systems by analyzing trends in the applicable previous systems.

    Guidance 4: Accelerate portions of the design to better inform the identification process.

    Guidance 5: The first key to overcoming the up-front investment is support from management.

    Guidance 6: Incentives can be used to encourage developers to make the up-front investment.

    Guidance 7: Assign responsibility for the identification of commonality.

Guidance 8: The extent to which an individual or group can identify opportunities for commonality is limited by their visibility.

Guidance 9: Evaluate monetary and non-monetary benefits and penalties.

Guidance 10: Give engineers visibility into the monetary effects of decisions and they will use that information to lower the system costs.

Guidance 11: Formal control processes to review and approve divergence will reduce unacceptable causes for divergence.

Guidance 12: Design with flexibility to allow a system to adapt to unanticipated changes.

Guidance 13: A development stream independent of a particular application ensures that development is not biased towards a specific application.

Guidance 14: Intelligent contract writing can be used to incentivize the contractor to seek out and implement commonality when beneficial.

Guidance 15: Contract writing can be used to avoid potential proprietary restrictions that have been witnessed in the past.

Guidance 16: When managing commonality between peer organizations, utilize standards.

Guidance 17: Reuse of software systems should be carefully tracked and tested.

Guidance 18: Software costs are more visible to developers and allow developers to consider economic consequences of designs.

Guidance 19: Service systems or other systems that are highly subject to customer demands should be flexible enough to adapt to changing needs.

Guidance 20: To maximize the life cycle benefits from commonality, create the larger volume, more capable system first.

- The management of commonality was observed in three different organizational and development scenarios:
  1. In-house single organization development
  2. In-house multiple organizations development
  3. External/contracted single organization

However the most difficult managerial case was not observed in the case studies, in which development of commonality must be managed between multiple contractors, as seen in Table 28.

Table 28:Development method and organizational structure of each of the case study subjects

	In-house	**Contracted**
Single-project	Core Flight Software System (Easiest)	Constellation Space Suit System
Peer Organizations	International Standard Payload Rack	(Hardest)

- Evaluating commonality as a Real Option allows developers to value the benefits of flexibility offered to the system. In addition, the decision analysis method used in the case study allows for the creation of a strategic development plan.

- The option to either use the previously developed components or discontinue common development is most valuable when there is a large amount of uncertainty in the system and on the benefits of commonality.

- The *cost of common development* up-front investment to develop components with the intent to become common may not have to be paid completely up-front. Economic analysis of the CSSS case study, in Chapter 7, showed that increased costs in common development, compared to independent development, were only incurred during the first three years of the life cycle and were completely recovered in the fourth year. The resultant up-front investment of common development was only \$6.4 million although the total development cost of system A was \$11 million higher than the independently developed system A.

8.3 Future Research

While the presented research does much to investigate the management of commonality within NASA systems, additional research should be conducted in order to further the understanding of commonality management and verify proposed guidance on an actual system. Proposed future research is discussed below.

- **Application of proposed guidance to a system that is attempting to apply commonality** in order to verify the described management guidance. The guidance described in Chapter 6 was created from the NASA case study findings, however in order to prove that the guidance represents the best management approach the guidance should be tested in the development of a common system.

- **Further development of the options-based economic model** to additionally consider economic revenue and additional options and uncertainties in common systems. Research should also be conducted to evaluate the potential of incorporating non-monetary effects of commonality into the model.

- **Further research on the existence and predictability of divergence in a common system.** Currently, the amount of divergence that will occur during development is completely unknown and unpredictable. Research should be conducted to determine in what situations divergence occurs and whether or not divergence can be predicted.

- **An additional case study should be conducted to capture lessons from the "hardest case."** The three NASA case studies conducted present a diverse view into NASA systems, varying in system type, development method, and organizational structure. However, the most difficult scenario to implement commonality was not observed, in which forward commonality is developed by multiple contractors, externally. Therefore an additional case study should be conducted to identify the challenges and best practices in the hardest category, as shown in Table 28.

- **Integration of the presented management guidance and the technical identification tools** that were developed by Hofstetter (2009) in order to form a more complete toolkit for the management and application of commonality that could be applied to future projects.

Bibliography

Amos, J. (2009-29-August). Europe looks to by Soyus craft. *BBC News* . London.

Baldwin, C. Y., & Clark, K. B. (2000). *Design Rules: The power of modularity.* Cambridge: The MIT Press.

Boas, R. C. (2008). Commonality in Complex Product Families: Implications of Divergence and Lifecycle Offsets. *Massachusetts Institute of Technology* .

Bush, P. G. (2004-14-January). *The White House.* Retrieved 2008-20-December from http://www.whitehouse.gov/news/releases/2004/01/20040114-3.html

Coan, D. A., & Bell, E. R. (2006). Essential Commonality for Effective Future Extravehicular Activity Operations. *Space Ops Conference Proceedings.* American Institute of Aeronautics and Astronautics.

Crites, J., & Tremblay, P. (1989). *Space Station Maintainability Design Requirements for Life Cycle Costs (Commonality & Standardization).* Hamilton Standard. NTRS.

Dahmus, J. B., Gonzalez-Zugasti, J. P., & Otto, K. N. (2001). Modular product architecture. *Design Studies* , *22*, 409-424.

de Neufville, R. (1990). *Applied Systems Analysis: Engineering Planning and Technology Management.* McGraw Hill Inc.

de Neufville, R. (n.d.). *Engineer Systems Analysis for Design.* Retrieved 2009-10 from http://ardent.mit.edu/real_options/Common_course_materials/slides.html

de Weck, O., de Neufville, R., & Chaize, M. (2004). Staged Deployment of Communications Satellite Constellations in Low Earth Orbit. *Journal of Aerospace Computing, Information, and Communication* , *1*, 119-136.

Desmond, J. P. (2008-November). Innovation Alive and Well. *Software Mag* .

ESAS, N. (2005). *NASA's Exploration Systems Architecture Study (ESAS).* NASA.

Federal Acquisition Regulation. (n.d.). Retrieved 2009-15-2 from Acquisition Central: http://www.acquisition.gov/far/

Fedor, W., Waiss, R., & Baune, M. *International commonality for Space Station.* Boeing Aerosopace Company, MBB/ERNO.

Fricke, E., & Schulz, A. P. (2005). Design for Changeability (DfC): Principles to Enable Changes in Systems Throughout Their Entire Lifecycle. *Systems Engineering* .

Gonzalez-Zugasti, J. P., Otto, K. N., & Baker, J. D. (2000). A Method for Architecting Product Platforms. *Research in Engineering Design* .

Gonzalez-Zugasti, J. P., Otto, K. N., & Whitcomb, C. A. (2007). Options-Based Multi-Objective Evaluation of Product Platforms. *Naval Engineers Journal* , 89-106.

Harland, D. M., & Catchpole, J. E. (2002). *Creating the International Space Station.* (J. Mason, Ed.) New York: Springer Praxis.

Hashimoto, H., Fukatsu, T., Ano, Y., Mizuno, M., Hashimoto, T., & Tokumura, T. (1998). Development of International Standard Payload Rack Structure for Space Station Science Operations. *American Institute of Aeronautics and Astronautics* .

Hofstetter, W. (2009). A Framework for Architecting Aerospace Portfolios with Commonality. MIT.

Hofstetter, W., de Weck, O., & Crawley, E. (2005). Modular Building Blocks for Manned Spacecraft: A Case Study for Moon and Mars Landing Systems. *INCOSE.*

Jordan, N. C. (2006). Multidisciplinary spacesuit modeling and optimization: requirement changes and recommendations for the next-generation spacsuit design. Massachusetts Institute of Technology.

Leveson, N. G., & Weiss, K. A. (2004). Making Embedded Software Reuse Practical and Safe. *SIGSOFT '04/FSE-12.* Newport Beach: ACM.

Leveson, N. (n.d.). Software and System Safety Research Group: A White Paper.

Leveson, N. (2008-4-12). System safety. *Notes from Class Presentation* .

Maier, M. W., & Rechtin, E. (2002). *The Art of Systems Architecting* (2nd Edition ed.). Boca Raton, Florida: CRC Press.

Meyer, M. H., & Seliger, R. (1998-Fall). Product Platforms in Software Development. *Sloan Management Review* .

(2008). *NASA Cost Estimating Handbook.* NASA. NASA.

NASA. (n.d.). *Interface Requirements Document - International Standard Payload Rack.*

NASA. (n.d.). *NASA - Partners Sign ISS Agreements.* Retrieved 2009-August from NASA.gov: http://www.nasa.gov/mission_pages/station/structure/elements/partners_agreement.html

NASA. (2008). *NASA Awards Contract for the Constellation Space Suit* . Retrieved 2008-10 from http://www.nasa.gov/home/hqnews/2008/jun/HQ_C08037_Constellation_Spacesuit.html

NASA, CSSS RFP . (n.d.). CSSS RFP, Attachment J-2.

NASA, CxP 70000. (n.d.). Constellation Architecture Requirements Document (CARD).

NASA, ESPO. (2007). RFP, Constellation Space Suit System (CSSS).

Neely, J. E., & de Neufville, R. (2001-2-November). Hybrid Real Options Valuation of Risky Product Development Projects. *International Technology, Policy, and Management* .

Newman, D., Eschenback, T., & Lavelle, J. (2004). *Engineering Economic Analysis.* New York: Oxford University Press.

Nilchiani, R., & Hastings, D. E. (2007). Measuring the Value of Flexibility in Space Systems: A Six-Element Framework. *Systems Engineering* , *10* (1), 26-44.

Quinn, S. M. (2008). Commonality of Ground Systems in Launch Operations. MIT.

Review of U.S. Human Spaceflight Plans Committee . (2009). *Final Report.* NASA.

Robertson, D., & Ulrich, K. (1998). Planning for Product Platforms. *Sloan Management Review* .

Robinson, J. A., Thumm, T. L., & Thomas, D. A. (2007). NASA utilization of the International Space Station and the Vision for Space Exploration. *Acta Astronautica* , *61*, 176-184.

Schulz, A. P., Clausing, D. P., Negele, H., & Fricke, E. (1999). Shifting the View in Systems Development p Technology Development at the Fuzzy Front End as a Key to Success. *Proceedings of 1999 ASME DETC.* Las Vegas: ASME.

Siddiqi, A., & de Weck, O. L. (2007). Spare Parts Requiremets for Space Missions with Reconfigurability and Commonality. *Journal of Spacecraft and Rockets* , *44* (1).

Sledd, A. M. (2000). The ISS EXPRESS Rack: An Innovative Approach for Rapid Integration. *Space Technology and Applications International Forum* (pp. 21-26). American Institute of Physics.

Stone, R. B., Wood, K. L., & Crawford, R. H. (2000). A Heuristic Method for Identifying Modules for Product Architectures. *Design Studies* , *21* (1), 5-31.

Thomas, K. S., & Mcmann, H. J. (2006). *US Spacesuits.* Chichester, UK: Springer/Praxis.

Uri, J. J. (2007). NASA Utilization of ISS - Past, Present and Future. *Bremen Microgravity Sci. Technol.* , *XIX* (5/6), 37-41.

Waiss, R. (1987). Cost Reduction on Large Space Systems Through Commonality. *American Institute of Aeronautics and Astronautics*, (p. 1).

Wilmot, J. (2008). A Core Plug and Play Architecture for Reusable Flight Software Systems.

Wilmot, J. (n.d.). Core Flight Software System (CFS) Presentation.

Yearson, J. D. (2009). *Product Family Design Using Smart Pareto Filters.* Thesis, Brigham Young University.

Appendix A: List of Acronyms

ACES	Advanced Crew Escape Suit
APAS	Androgynous Peripheral Attach System
ASK	Academy Sharing Knowledge
ATA	Architecture Trade Analysis
BSC	Body Seal Closure
CARD	Constellation Architecture Requirements Document
CBM	Common Berthing Mechanism
CEI	Contract End Item
cFE	core Flight Executive
CFS	Core Flight-Software System
COTS	Commercial Orbital Transportation Services
CSA	Canadian Space Agency
CSSS	Constellation Space Suit System
CTSD	Crew and Thermal Systems Division
CxP	Constellation Program
D&C	Design and Construction Standards
DDT&E	Design, Development, Testing, and Evaluation
DfC	Design for Changeability
DoD	Department of Defense
ECLSS	Environmental Control and Life Support System
EDS	Earth Departure Stage
ESA	European Space Agency
ESAS	Exploration System Architecture Study
ESEP	EVA Systems Engineering Panel
EMU	Extravehicular Mobility Unit
ESAS	Exploration Systems Architecture Study
ESPO	EVA System Project Office
EVA	Extra-Vehicular Activity

EXPRESS	EXpediting the PRocessing of Experiments to the Space Station
FAR	Federal Acquisition Regulation
FOM	Figure of Merit
FSW	Flight Software
GSFC	Goddard Space Flight Center
HUT	Hard Upper Torso
ICD	Interface Control Document
ICWG	Interface Control Working Group
IDIQ	Indefinite Delivery/Indefinite Quantity
IGA	Intergovernmental Agreement
IC	Initial Capability
ISPR	International Standard Payload Rack
ISS	International Space Station
JAXA	Japanese Aerospace Exploration Agency
JSC	Johnson Space Center
JSF	Joint Strike Fighter
LEA	Launch, Entry, and Abort
LEO	Low-Earth Orbit
LIDS	Low Impact Docking System
MOU	Memorandum of Understanding
MPLM	Multi-Purpose Logistics Module
NAFCOM	NASA/Air Force Cost Model
NASA	National Aeronautics and Space Administration
NPV	Net Present Value
OSAL	Operating System Abstraction Layer
PCB	Project Control Board
PRN	Program Release Notice
RCA	Rapid Cycle Amine
RFI	Request for Information
RFP	Request for Proposals
RKA	Russian Federal Space Agency

ROI	Return on Investment
SE&I	Systems Engineering and Integration
SIG	System Integration Group
SOM	System Overlap Matrix
SRD	System Requirements Document
TMG	Thermal and Micrometeorite Garment
TPM	Technical Performance Metric
VIE	Vehicle Interface Element
VSE	Vision for Space Exploration

Appendix B: Management Guidance

Guidance 1: The identification process must begin early in development in order to maximize the potential benefits.

Guidance 2: Design with flexibility to reduce the effect of uncertainty on the system.

Guidance 3: Reduce uncertainty in future systems by analyzing trends in the applicable previous systems.

Guidance 4: Accelerate portions of the design to better inform the identification process.

Guidance 5: The first key to overcoming the up-front investment is support from management.

Guidance 6: Incentives can be used to encourage developers to make the up-front investment.

Guidance 7: Assign responsibility for the identification of commonality.

Guidance 8: The extent to which an individual or group can identify opportunities for commonality is limited by their visibility.

Guidance 9: Evaluate monetary and non-monetary benefits and penalties.

Guidance 10: Give engineers visibility into the monetary effects of decisions and they will use that information to lower the system costs.

Guidance 11: Formal control processes to review and approve divergence will reduce unacceptable causes for divergence.

Guidance 12: Design with flexibility to allow a system to adapt to unanticipated changes.

Guidance 13: A development stream independent of a particular application ensures that development is not biased towards a specific application.

Guidance 14: Intelligent contract writing can be used to incentivize the contractor to seek out and implement commonality when beneficial.

Guidance 15: Contract writing can be used to avoid potential proprietary restrictions that have been witnessed in the past.

Guidance 16: When managing commonality between peer organizations, utilize standards.

Guidance 17: Reuse of software systems should be carefully tracked and tested.

Guidance 18: Software costs are more visible to developers and allow developers to consider economic consequences of designs.

Guidance 19: Service systems or other systems that are highly subject to customer demands should be flexible enough to adapt to changing needs.

Guidance 20: To maximize the life cycle benefits from commonality, create the larger volume, more capable system first.